

(19) 日本国特許庁 (JP)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2004-504681

(P2004-504681A)

(43) 公表日 平成16年2月12日 (2004.2.12)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
G06F 12/00	G06F 12/00 546K	5B056
G06F 13/00	G06F 13/00 540B	5B082
G06F 17/18	G06F 17/18 Z	

審査請求 未請求 予備審査請求 有 (全 67 頁)

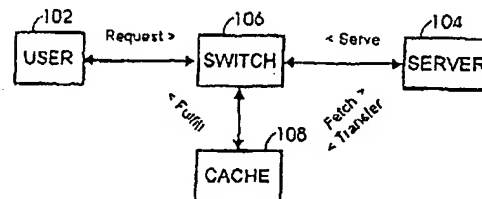
(21) 出願番号	特願2002-513142 (P2002-513142)	(71) 出願人	503026314
(86) (22) 出願日	平成13年7月16日 (2001.7.16)		エプ리케이션 ネットワークス リミテッド
(85) 翻訳文提出日	平成15年1月17日 (2003.1.17)		EPLICATION NETWORKS LTD.
(86) 国際出願番号	PCT/IL2001/000651		イスラエル国、ホド ハシャロン 452
(87) 国際公開番号	W02002/007364		41、ハナガル ストリート 5ビー
(87) 国際公開日	平成14年1月24日 (2002.1.24)		5B Hanagar St., 4524
(31) 優先権主張番号	60/218,559		1 Hod Hasharon, Israel
(32) 優先日	平成12年7月17日 (2000.7.17)	(74) 代理人	100092897
(33) 優先権主張国	米国 (US)		弁理士 大西 正悟
		(74) 代理人	100115200
			弁理士 山口 修之

最終頁に続く

(54) 【発明の名称】 キャッシュ分散ネットワーク上に静的及び動的コンテンツをキャッシュすることによる、WWWサイトのビット・レート節約及びQoS改善

(57) 【要約】

動的コンテンツを持つオブジェクトのコピーを保存及び更新するために、インターネット内にキャッシュを配置する。オブジェクトの更新特性を判定し、この更新特性に基づいてオブジェクトに対する残存時間 (TTL) パラメータを調整する。一般的に、TTLが経過時間よりも少ない場合、オブジェクトを更新する。オブジェクトに対するTTLは、(i) エラーの確率を所定の誤り確率閾値よりも少なく維持する、(ii) 誤り率を所定の誤り確率閾値よりも少なく維持する、あるいは (iii) 遅延時間を所定の遅延閾値よりも少なく維持するように調整する。キャッシュは、専用のマシンで、ウェブ・ブラウジングが、本来のサイトへ向かって全経路を通過する代わりにキャッシュを通過するよう、多くの異なるロケーション内に、理想的には、ロケーション内の最大数のユーザへインターネット接続を提供するISPのネットワーク内に配置することが好ましい。このようにすれば、通信の大部分がISPの内部ネットワーク内、あるいはその近くに留まり、負荷が大きいインターネット・バックボーンを通る必要がないため、これらのISP



BEST AVAILABLE COPY

【特許請求の範囲】

【請求項 1】

インターネット上の動的コンテンツを持つオブジェクトをキャッシュするためのシステムであって、

前記インターネットに接続され、動的コンテンツのコピーを保存及び更新するためのキャッシュ、

前記オブジェクトの一つ以上をモニターし、その統計的更新特性を判定するための手段、そして

前記統計的更新特性に基づいて前記オブジェクトに対する残存時間 (T T L) パラメータを調節するための手段からなるシステム。

10

【請求項 2】

前記統計的更新特性が変化統計値を含む、請求項 1 に記載のシステム。

【請求項 3】

前記統計的更新特性がアクセス統計値を含む、請求項 1 に記載のシステム。

【請求項 4】

前記キャッシュが、主要なインターネット・スイッチングロケーション、ダイヤル・イン集合ポイント、そして企業ゲートウェイからなるグループから選択したロケーションに配置される、請求項 1 に記載のシステム。

【請求項 5】

前記キャッシュが、また、静的コンテンツのコピーをも保存し更新する、請求項 1 に記載のシステム。

20

【請求項 6】

さらに、オブジェクトの経過時間を判定するための手段、そして

前記オブジェクトに対する T T L が、経過時間よりも少ない場合に前記オブジェクトを更新する手段からなる、請求項 1 に記載のシステム。

【請求項 7】

各オブジェクトに対する前記 T T L が、前記オブジェクトの変化の平均時間に応じて計算される、請求項 6 に記載のシステム。

【請求項 8】

さらに、前記オブジェクトの各々に対する誤り確率を判定するための手段、そしてその誤り確率を所定の誤り確率閾値よりも低く維持するために、前記オブジェクトの各々に対する T T L を調節するための手段からなる、請求項 1 に記載のシステム。

30

【請求項 9】

さらに、前記オブジェクトの各々に対する誤り率を判定するための手段、そしてその誤り率を所定の誤り確率閾値よりも低く維持するために、前記オブジェクトの各々に対する T T L を調節するための手段からなる、請求項 1 に記載のシステム。

【請求項 10】

さらに、前記オブジェクトの各々の遅延時間を判定するための手段、そして

その遅延時間を所定の遅延閾値よりも少なく維持するために、前記オブジェクトの各々に対する T T L を調節するための手段からなる、請求項 1 に記載のシステム。

40

【請求項 11】

さらに、前記オブジェクトの各々に対する誤り確率、誤り率、そして遅延時間からなるグループから選択した少なくとも一つのオブジェクト特性を判定するための手段、そして前記選択オブジェクト特性を各々の閾値よりも低く維持するために、T T L を調節するための手段からなる、請求項 1 に記載のシステム。

【請求項 12】

さらに、前記オブジェクトの各々に対する T T L の調整を、所定の最小値 (T m i n) 及び最大値 (T m a x) に囲まれる T T L の範囲内に制限するための手段からなる、請求項 11 に記載のシステム。

【請求項 13】

50

インターネット上の動的コンテンツを持つオブジェクトをキャッシュするためのシステムであって、

動的コンテンツのコピーを保存及び更新するための、前記インターネットに接続されたキャッシュ、

前記オブジェクトの一つ以上をモニターし、その更新特性を判定するための手段、

前記更新特性に基づいて前記オブジェクトに対する残存時間 (T T L) パラメータを調節するための手段、

前記オブジェクトの各々に対する誤り確率を判定するための手段、そして

その誤り確率を所定の誤り確率閾値よりも低く維持するために、前記オブジェクトの各々に対する前記 T T L を調節するための手段からなり、

各オブジェクトに対する前記 T T L が、次の方程式に従って計算され、

$$T \leq (W + \tau \rho_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \rho_0} \right)^2} \right\}$$

前記式において、

T は動的オブジェクトに対する残存時間で、

W は、本来のオブジェクトが変化してからの任意の時間 (すなわち、どれほど古いか) で

、
τ は、前記オブジェクトの変化統計値から判定される、変化間の平均時間で、

ρ₀ は、任意の時間 W を超える期間に、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした要求数の、全要求数に対する平均比率である最大誤り確率であることを特徴とするシステム。

【請求項 14】

インターネット上の動的コンテンツを持つオブジェクトをキャッシュするためのシステムであって、

前記インターネットに接続され、動的コンテンツのコピーを保存し更新するためのキャッシュ、

前記オブジェクトの一つ以上をモニターし、その更新特性を判定するための手段、

前記更新特性に基づいて前記オブジェクトに対する残存時間 (T T L) パラメータを調節するための手段、

前記オブジェクトの各々に対する誤り率を判定するための手段、そして

その誤り率を所定の誤り確率閾値よりも低く維持するために、前記オブジェクトの各々に対する T T L を調節するための手段からなり、

各オブジェクトに対する T T L が、次の方程式に従って計算され、

$$T \leq \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

前記式において、

T は動的オブジェクトに対する残存時間で、

W は、本来のオブジェクトが変化してからの任意の時間 (すなわち、どれほど古いか) で

、
τ は、前記オブジェクトの変化統計値から判定される、変化間の平均時間で、

n は単位時間当たりのユーザ要求数 (例えば、頻度) で、

n₀ は、任意の時間 W を超える期間において、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした単位時間当たりの平均要求数である最大誤り率であることを特徴とするシステム。

【請求項 15】

インターネット上の動的コンテンツを持つオブジェクトをキャッシュするためのシステムであって、

前記インターネットに接続され、動的コンテンツのコピーを保存及び更新するためのキャッシュ、

前記オブジェクトの一つ以上をモニターし、その更新特性を判定するための手段、

前記更新特性に基づいて前記オブジェクトに対する残存時間 (T T L) パラメータを調節するための手段、

前記オブジェクトの各々に対する遅延時間を判定するための手段、そして

その遅延時間を所定の遅延閾値よりも低く維持するために、前記オブジェクトの各々に対する T T L を調節するための手段からなり、 10

各オブジェクトに対する T T L は、次の方程式に従って計算され、

$$T \leq \frac{6D_0}{1 + 3 \frac{D_0}{\tau}}$$

前記式において、

T は動的オブジェクトに対する残存時間で、

τ は、前記オブジェクトの変化統計値から判定される、変化間の平均時間で、

D_0 は、オブジェクト変化と、キャッシュ・コピーのリフレッシュとの間の平均時間である最大遅延であることを特徴とするシステム。 20

【請求項 16】

動的コンテンツを持つ情報に対するユーザ要求に応答するための方法であって、

前記動的コンテンツのコピーをキャッシュに保存すること、

前記動的コンテンツに対する残存時間 (T T L) を設定すること、

前記キャッシュにおいて前記ユーザ要求を受信すること、

前記 T T L がその経過時間よりも多い場合は、前記動的コンテンツの保存コピーで前記ユーザ要求に応答すること、

前記保存コピーの T T L がその経過時間よりも少ない場合は、前記動的コンテンツの更新コピーを検索して、その更新コピーで前記ユーザ要求に応答すること、 30

前記動的コンテンツに対する平均更新頻度を判定すること、そして

その平均更新頻度の関数として前記動的コンテンツに対する前記 T T L を決定することからなる方法。

【請求項 17】

さらに、前記キャッシュに静的コンテンツのコピーを保存することからなる、請求項 16 に記載の方法。

【請求項 18】

前記情報がウェブ・ページとして表され、さらに、ユーザが初めて前記ウェブ・ページを受信するときに、前記ウェブ・ページが前記ユーザのブラウザ内にクッキーを植えつけることからなる、請求項 16 に記載の方法。 40

【請求項 19】

その後、前記末端ユーザが前記ウェブ・ページを求めるときはいつも、前記ブラウザが前記クッキーを送る、請求項 18 に記載の方法。

【請求項 20】

さらに、前記キャッシュが、前記クッキーを調べ、前記要求のウェブ・ページが単一末端ユーザの個人的なページであるかどうかを見分けることからなる、請求項 19 に記載の方法。

【請求項 21】

さらに、前記要求のウェブ・ページが単一末端ユーザの個人的なウェブ・ページである場合は、前記ウェブ・ページがキャッシュされないことからなる、請求項 18 に記載の方法 50

【請求項 22】

さらに、前記情報が、アクセスの度に変更されることをサポートしている場合は、 $TTL = 0$ を設定することからなる、請求項 16 に記載の方法。

【請求項 23】

動的コンテンツを持つ情報に対するユーザ要求に応答するための方法であって、
 前記動的コンテンツのコピーをキャッシュに保存すること、
 前記動的コンテンツに対する残存時間 (TTL) を設定すること、
 前記キャッシュにおいてユーザ要求を受信すること、
 前記 TTL がその経過時間よりも多い場合は、前記動的コンテンツの保存コピーで前記ユーザ要求に応答すること、
 前記保存コピーの TTL がその経過時間よりも少ない場合は、前記動的コンテンツの更新コピーを検索して、この更新コピーで前記ユーザ要求に応答すること、
 前記動的コンテンツに対する平均更新頻度を判定すること、そして
 その平均更新頻度と所定の誤り確率閾値との関数として前記動的コンテンツに対する TTL を決定することからなる方法。

【請求項 24】

さらに、前記動的コンテンツに対するユーザ要求の頻度に従って、前記動的コンテンツに対する TTL を調節することからなる、請求項 23 に記載の方法。

【請求項 25】

動的コンテンツを持つ情報に対するユーザ要求に応答するための方法であって、
 前記動的コンテンツのコピーをキャッシュ内に保存すること、
 前記動的コンテンツに対する残存時間 (TTL) を設定すること、
 前記ユーザ要求を前記キャッシュにおいて受信すること、
 前記 TTL がその経過時間よりも多い場合は、前記動的コンテンツの保存コピーで前記ユーザ要求に応答すること、そして
 前記保存コピーの TTL がその経過時間より少ない場合は、前記動的コンテンツの更新コピーを検索して、その更新コピーで前記ユーザ要求に応答することからなり、
 各オブジェクトに対する前記 TTL は、次の方程式に従って計算され、

$$T \leq (W + \tau p_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right\}$$

前記式において、

T は動的オブジェクトに対する残存時間で、

W は、本来のオブジェクトが変化してからの任意の時間（すなわち、どれほど古いか）で

τ は、前記オブジェクトの変化統計値から判定される、変化間の平均時間で、

p_0 は、任意の時間 W を超える期間に、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした要求数の、全要求数に対する平均比率である最大誤り確率であることを特徴とする方法。

【請求項 26】

動的コンテンツを持つ情報に対するユーザ要求に応答するための方法であって、
 前記動的コンテンツのコピーをキャッシュ内に保存すること、
 前記動的コンテンツに対する残存時間 (TTL) を設定すること、
 前記キャッシュにおいて前記ユーザ要求を受信すること、
 前記 TTL がその経過時間よりも多い場合は、前記動的コンテンツの保存コピーで前記ユーザ要求に応答すること、そして
 前記保存コピーの TTL がその経過時間より少ない場合は、前記動的コンテンツの更新コピーを検索して、その更新コピーで前記ユーザ要求に応答することからなり、

各オブジェクトに対する T T L が、次の方程式に従って計算され、

$$T \leq \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

前記式において、

T は動的オブジェクトに対する残存時間で、

W は、本来のオブジェクトが変化してからの任意の時間（すなわち、どれほど古いか）で 10

、
τ は、前記オブジェクトの変化統計値から判定される、変化間の平均時間で、

n は単位時間当たりのユーザ要求数（例えば、頻度）で、

n₀ は、任意の時間 W を超える期間において、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした単位時間当たりの平均要求数である最大誤り率であることを特徴とする方法。

【請求項 27】

動的コンテンツを持つ情報に対するユーザ要求に応答するための方法であって、

前記動的コンテンツのコピーをキャッシュに保存すること、

前記動的コンテンツに対する残存時間（T T L）を設定すること、 20

前記キャッシュにおいて前記ユーザ要求を受信すること、

前記 T T L がその経過時間よりも多い場合は、前記動的コンテンツの保存コピーで前記ユーザ要求に応答すること、そして

前記保存コピーの T T L がその経過時間よりも少ない場合は、前記動的コンテンツの更新コピーを検索して、その更新コピーで前記ユーザ要求に応答することからなり、

各オブジェクトに対する T T L は、次の方程式に従って計算され、

$$T \leq \frac{6D_0}{1 + 3 \frac{D_0}{\tau}}$$

30

前記式において、

T は動的オブジェクトに対する残存時間で、

τ は、前記オブジェクトの変化統計値から判定される、変化間の平均時間で、

D₀ は、オブジェクト変化と、キャッシュ・コピーのリフレッシュとの間の平均時間である最大遅延であることを特徴とする方法。

【請求項 28】

サーバから生じる、動的コンテンツを持つオブジェクトに対するユーザ要求に応答するための方法であって、

前記オブジェクトのコピーをキャッシュに保存すること、

前記オブジェクトに対する残存時間（T T L）を設定すること、 40

前記キャッシュにおいて前記ユーザ要求を受信すること、

前記 T T L がその経過時間よりも多い場合は、前記オブジェクトの保存コピーで前記ユーザ要求を満たすこと、

前記保存コピーの T T L がその経過時間よりも少ない場合は、前記オブジェクトの更新コピーをフェッチして、その更新コピーで前記ユーザ要求に応答すること、そして

前記キャッシュにおいて、

まず、前記オブジェクトに対する T T L を妥当な下限（T m i n）に設定すること、そして

前記オブジェクトが実際に変化する頻度に基づいて前記オブジェクトに対する T T L を調節することを特徴とする方法。 50

【請求項 29】

情報がウェブ・ページとして表され、さらに、ユーザが初めて前記ウェブ・ページを受信するときに、前記ウェブ・ページが前記ユーザのブラウザ内にクッキーを植えつけることからなる、請求項 28 に記載の方法。

【請求項 30】

その後、前記末端ユーザが前記ウェブ・ページを求めるときはいつも、前記ブラウザが前記クッキーを送る、請求項 29 に記載の方法。

【請求項 31】

さらに、前記キャッシュが、前記クッキーを調べ、前記要求のウェブ・ページが単一末端ユーザの個人的なページであるかどうかを見分けることからなる、請求項 30 に記載の方法。

10

【請求項 32】

さらに、前記要求のウェブ・ページが単一末端ユーザの個人的なウェブ・ページである場合は、前記ウェブ・ページがキャッシュされないことからなる、請求項 29 に記載の方法。

【請求項 33】

さらに、前記情報が、アクセスの度に変更されることをサポートしている場合は、 $TTL = 0$ を設定することからなる、請求項 28 に記載の方法。

【請求項 34】

サーバから生じる、動的コンテンツを持つオブジェクトに対するユーザ要求に応答するための方法であって、

20

前記オブジェクトのコピーをキャッシュ内に保存すること、

前記オブジェクトに対する残存時間 (TTL) を設定すること、

前記キャッシュにおいて前記ユーザ要求を受信すること、

前記 TTL がその経過時間よりも多い場合は、前記オブジェクトの保存コピーで前記ユーザ要求を満たすこと、

前記保存コピーの TTL がその経過時間よりも少ない場合は、前記オブジェクトの更新コピーをフェッチして、その更新コピーで前記ユーザ要求に応答すること、そして

前記キャッシュが前記サーバから前記オブジェクトをフェッチする度に、次の処理を行うこと、

30

a. 同じオブジェクトに対するもう一つのフェッチが進行中である場合は、その先行するフェッチが完了するのを待つこと、

b. 前記サーバから前記オブジェクトをフェッチすること、

c. 前記キャッシュ内のコピーを、もしそれが存在するならば、最後にフェッチして以来前記オブジェクトが変化したかどうかを判定するために比較した後、前記フェッチしたオブジェクトで置き換えること、

d. 前記オブジェクトの変化統計値を初期化する、あるいは相応に更新すること、

e. 前記サーバの応答に応じて、前記オブジェクトに静的あるいは動的コンテンツと標識すること、そして

f. 前記オブジェクトが動的である場合は、前記オブジェクトの変化統計値から判定した変化 (r) 間の平均時間、前記オブジェクトのアクセス統計値から判定した単位時間ユーザ要求数 (n)、そして次の処理 (A-E) の一つに応じて、前記 TTL (T) を適当な値に設定することからなり、

40

A は最大誤り確率、

B は最大誤り率、

C は最大遅延、

D は T が最低になる上記 (A、B、C) の組合せ、

E は T_{min} と T_{max} とが挟む「妥当な」値の所定範囲内に T を留める上記 (A、B、C) の組合せであることを特徴とする方法。

【発明の詳細な説明】

50

【0001】

【発明の技術的な分野】

本発明は、インターネット上の情報の保存及び通信、特に、インターネット上のウェブページ・コンテンツをキャッシュすることに関する。

【0002】

【関連出願との相互参照】

これは、2000年7月17日に出願した、特許出願中の仮特許出願第60/218,559号を、共通所有で継続するものである。

【0003】

【発明の背景】

リソースを共有できるように、二つ以上のコンピュータを接続したものを、「ネットワーク」と呼ぶ。二つ以上のネットワークを結合したものを、「インターネット」と呼ぶ。「インターネット」は、複数のネットワークを相互に連結し、すべてにTCP/IP（伝送制御プロトコル／インターネット・プロトコル）として知られる共通のプロトコルを用いるグローバルなインターネットである。

【0004】

ワールド・ワイド・ウェブ（「WWW」、あるいは単に「ウェブ」）は、コンピュータ、特に、インターネット上のハイパーテキスト（HTTP）サーバ間の情報交換を可能にする規格及びプロトコルのセットで、インタラクティブなマルチメディアのリソースの巨大なコレクションを構築するものである。インターネットの通信は、ピアリングと呼ぶシステムの下で互いに信号を転送するUUNetあるいはDigex等の、多数の「バックボーン」キャリアを通過する。

【0005】

「サービスの品質」（QoS）は、コンテンツ・ソース（通常、サーバ、あるいはインターネット・サービス・プロバイダ、ISP）から、ユーザのコンピュータ（通常、コンテンツを見るために用いる「ブラウザ」ソフトウェアを走らせている）へ情報が移送される速度及び信頼性の測度である。「ビット・レート」は、任意の時間におけるネットワーク（あるいは、その任意の一部）上を通過する情報量の測度である。ビット・レートが高ければ、それだけより多くのデータが移送できる。待ち時間は、ソースからユーザへ、データの packets がインターネットを通過する時間の測度である。

【0006】

新しく資本的資源を注入することによってビット・レートを増加させることができるが、ビット・レートの増加は、単に待ち時間を間接的に処理するもので、ビット・レートの増加によって、過負荷の、あるいは元々遅いサーバを速めることはできない。これらの障壁に打ち勝つためには他の解決方法が必要である。一つの解決方法として、コンテンツをキャッシュすることが、1999年3月のマイケル・A・ゲールド氏による、「ウェブ・コンテンツ供給を最適化するネットワーク・キャッシュ・ガイド」という題名の記事に詳細に論じられている。この記事は、次のアドレスから（インターネットを介して）オンラインで利用可能である。

http://www.inktomil.com/products/network/tech_resources/tech/cachingguide.pdf

そのテキスト・バージョンは、次のアドレスからオンラインで利用可能である。

http://www.google.com/search?q=cache:www.inktomil.com/products/network/tech_resources/tech/cachingguide.pdf

【0007】

ゲールド氏の記事に次のように述べられている。

【0008】

「性能を改善して、より予測可能にする一つの方法は、コンテンツ・ソースからユーザのブラウザまでの、インターネット上の長距離通信によって導入されるばらつきを最小にす

10

20

30

40

50

ることである。アクセスが頻繁なコンテンツを、ユーザにより近い箇所に保存することによって、インターネットにおける多くの待ち時間及び予想不可能な遅延を解消することができる。この技術は、キャッシングと呼ばれる。キャッシングは、ウェブ・サーバからのコンテンツ・オブジェクトを、より速く検索が可能のようにユーザにより近い所に保存する手段である。テキスト・ページ、イメージ、そして他のコンテンツを含むオブジェクトの倉庫は、ウェブ・キャッシュと呼ぶ。」(ゲールド、ページ2)

【0009】

キャッシングは、ユーザに対するQoSを改善するアプローチであり、また、ネットワーク・サービス・プロバイダに対する経済的な改善もある。アクセスが頻繁なウェブ・コンテンツをユーザの近くに保存することによって、遠隔なサイトに存在するコンテンツを検索するための、通過に必要な跳ぶ回数を減少させることができる。(ゲールド、ページ8)

10

【0010】

実際、コンテンツの要求は、ユーザのブラウザから生じて、まず、キャッシング・サーバに導かれる。コンテンツがキャッシュ内に現在含まれており、そして最新である場合は、ブラウザの要求を発信サーバへ送ることなく、そのコンテンツをユーザに送る。これによって、コンテンツの転送に対する待ち時間を減少させることができる。コンテンツをユーザへ転送するための時間が減少する。このことは、また、要求を満たすためにサービス・プロバイダがインターネットの上流から検索しなければならないデータの量を減少させる。(ゲールド、ページ8)

20

【0011】

キャッシングは、コンピュータ・ハードウェア・デザインの分野において、よく理解されているコンセプトである。モデム・マイクロプロセッサは、低速なメモリから読み取る代わりに、高速なメモリのオンチップ・キャッシュを用いて、頻繁に使用する命令及びデータを近くに保存する。コンピュータは、RAMキャッシュ、そしてオンディスク・キャッシュさえも含む、さらに複数のレベルのキャッシングを具現している。すべて、命令及びデータを読み取る待ち時間を減少させるようにデザインされ、システム内のデータの転送を速める。キャッシングの基本的な原理は、アクセスが頻繁なデータを、データの本来のロケーションよりも、より速くアクセスが可能なロケーションに保存することである。システムのCPUの場合には、本来のロケーションはディスクである。ウェブ上では、本来のロケーションはインターネット上の発信サーバである。(ゲールド、ページ8?9)

30

【0012】

ネットワーク・キャッシングは、新しい概念であるが、全く同じ原理に基づいており、ユーザの利用をコスト効果的に改善する同じ利点を持つ。(ゲールド、ページ3)

【0013】

「キャッシングは、ウェブ・ページのダウンロードの待ち時間(何かが始まる前に待たなければならない時間)を減少させる、そしてダウンロード・スピード(すべてのダウンロードが終了するための時間)がより速いという点でエンド・ユーザに直接的な利点を提供する。しかし、キャッシングは、ネットワーク・サービス・プロバイダにも利点を提供する。ユーザが遠隔のサーバから求めるウェブ・コンテンツを近くに保存する、すなわちキャッシュすることによって、インターネット上へ送らなければならない要求が減少する。その結果、キャッシュを維持しているアクセス・プロバイダは、上流のビット・レート要求が減少する。これは、満足がいくウェブ体験をその顧客に提供するためにサービス・プロバイダが購入しなければならないビット・レートを減少させる。(ゲールド、ページ3)

40

【0014】

「キャッシングは、また、多くの顧客のためにアクセス・プロバイダがウェブ・サイトを主催するウェブ・ホスティング環境において複数の利点を提供する。関心度が高いコンテンツが引き起こす通信スパイクの衝撃を減少させることができ、また、追加容量、サービス・レベルの保証、そして複製サービスを提供する等の、種々の付加価値サービスの基盤

50

として役立つ。(グールド、ページ3)

【0015】

「ネットワーク・データの[キャッシング]は、最適な体験と予測可能な応答時間とをユーザに提供することを可能にする。通常のキャッシュ・ヒット率で、ユーザは、高品質なサービス(QoS)を体験することができる。QoSの改善は、ISP、コンテンツ・プロバイダ、そして企業サイトにかなりの利点を提供する。より良いQoSは、顧客の関心を引きつけ、顧客の維持率を高める。したがって、アクセス・プロバイダ及びコンテンツ・プロバイダの、両方のブランドを高める。コンテンツ・プロバイダは、ネットワークを通じてコンテンツをユーザの近くに十分にキャッシュさせることを確実に行えば、究極的に、より多くのビジターがコンテンツにアクセスするようになる。(グールド、ページ5)

10

【0016】

「ウェブ・ブラウザも、キャッシングの形態を具現している。ユーザのハードディスク上に最近アクセスしたウェブ・コンテンツを保存し、インターネットにアクセスする代わりに、そのローカル・キャッシュからファイルを読み取る。これは、ユーザが、セッション中にブラウザ内の「バック」及び「フォワード」ボタンを叩いたときによく機能するが、ユーザがサイトを初めてチェックする場合には何もしない。(グールド、ページ9)

【0017】

「ブラウザのキャッシュ、そしてウェブ・サーバのキャッシュのいずれも、ネットワーク性能の問題を扱うことはできない。しかし、ユーザと発信ウェブ・サイトとの間のネットワーク上にウェブ・コンテンツのキャッシュを配置することによって、共通にアクセスされるコンテンツがインターネット上を移動しなければならない距離が減少するため、ユーザは、素早い反応、そして高速な性能を体験することができる。ネットワーク・キャッシングは、いくつかのコンテンツが他のコンテンツよりも頻繁にアクセスされるという事実を利用している。キャッシングによる解決を具現することによって、ネットワーク・サービス・プロバイダは、インターネットへの高ビット・レート接続のための全必要条件を減少させながら、顧客に対して、より良いウェブ体験を提供することができる。(グールド、ページ9)

20

【0018】

「キャッシング解決策がない場合、ブラウザからのコンテンツの要求と、発信サーバから供給されるコンテンツが、要求するコンピュータから、コンテンツを持つコンピュータまでの全行程を繰り返し通過しなくてはならない。ウェブ・ブラウザは、インターネット上の特定なサーバ上の特定なウェブ・ページを指名するユニフォーム・リソース・ロケータ(URL)の要求を送る。この要求は、通常のTCP/IPネットワーク通信を介してサーバへ経路を決められる。(HTTPサーバとしても知られる)ウェブ・サーバから求められるコンテンツは、グラフィックスを含む、一つ以上の追加ファイルへのリンクを持つ静的なHTMLページであるかもしれない。コンテンツは、また、サーチ・エンジン、データベース照会、あるいはウェブ・アプリケーションから生成される、動的に作成されるページであるかもしれない。HTTPサーバは、ウェブ・ブラウザへ、一度に一つのファイルずつ、求められたファイルを返す。動的に作成されるページでさえ、しばしば、最終的なページを作成するために、動的コンテンツに結合される静的な成分を持つ。(グールド、ページ13)

30

40

【0019】

「キャッシングを用いると、アクセスが頻繁なコンテンツは、ユーザの近くに保存される。ユーザの企業内LAN上のファイアウォール内に、ユーザのISPに、あるいは他のネットワーク・アクセス・ポイント(NAP)、または大多数のウェブ・サーバよりもユーザにより近く位置するプレゼンス・ポイント(POP)に保存してもよい。要求されたコンテンツあるいはファイルが、キャッシュ内に位置し、それが現在のものである、すなわち新鮮である場合は、これをキャッシュ・ヒットと見なす。これらのキャッシュから、より頻繁にユーザ要求が満たされる場合、ヒット率が高く、ユーザのパフォーマンスが良い

50

ため、サービス・プロバイダがユーザのためにインターネットから検索するデータは少ない。(ゲールド、ページ14)

【0020】

「同様の問題は、FTPクライアント・アプリケーションが提出するファイル要求の各々に対してFTPサーバが対処する、FTPファイル転送においても存在する。FTPが転送するファイルの通常のサイズは典型的なHTMLファイルよりも大きいため、FTPにおいては、遅延及びボトルネックはより大きな問題である。また、ストリーミングオーディオ及びビデオは、末端ユーザの近くにコンテンツをキャッシュすることから利益を受けることができるインターネット・アプリケーションのもう一つの例である。(ゲールド、ページ14)

10

【0021】

「ネットワーク・キャッシングは、多くの異なるプロトコルを介して供給されるコンテンツに応用できる。例えば、HTTP、NNTP、FTP、RTSPがある。すべては、いくつかの割合で静的コンテンツを持ち、そして高度な用途があるという特徴がある。もちろん、キャッシュ・サーバが各プロトコルをサポートすることが必要である。」(ページ14)

【0022】

「キャッシュを新鮮に保つ必要があるため、たとえユーザが求めるコンテンツの全ビットがキャッシュに存在するとしても、ISPからインターネットへの通信は発生する。ページの新鮮さを保証する必要がある、新しいコンテンツをダウンロードしなくてはならない。しかし、キャッシングを用いることによって、ビット・レートの利用は最適化される。動的コンテンツに対してさえも、キャッシュを用いることは可能である。なぜなら、これらのページは、キャッシュから利用できるいくつかの静的コンテンツを持つためである。通信の分布及びキャッシュの大きさに応じて、ユーザによるHTTP要求の最高40パーセントまでを、キャッシュを用いることによってネットワークから取り除くことができる。これにより、ネットワークをより効率的に保ち、低コストで、より良い性能を提供することができる。」(ゲールド、ページ14)

20

【0023】

ゲールド氏の記事は、キャッシュの有効な用途、負荷バランス、インフラ内のどこにキャッシュを設けるか、そしてキャッシュに効果的なウェブコンテンツのデザインに関して論じている。また、例えば、ウェブ・キャッシュ制御プロトコル(WCCP)等の開発済みプロトコルの言及もある(ページ18)。また、HTTPプロトコル内の「期限切れ」及び「最大期限」ヘッダーの適当な使用に関する議論もある。(ゲールド、ページ27)。そして、ゲールド氏は、特に次のことを力説している。

30

【0024】

「優れたデザインによって、動的に生成されるページでさえも、キャッシングから利益を得ることができる。ページ内の動的コンテンツを静的コンテンツから分離することによって、静的コンテンツはキャッシュし、動的コンテンツは別に検索してダウンロードすることができる。(ゲールド、ページ28、強調有り)

【0025】

したがって、専用コンピュータのラム及びハードディスクを用いて、ウェブ・サイトから静的データをキャッシュし、このコンピュータを、WWW(「ウェブ」)サーバとユーザとの間のプロキシとして機能させることが知られている。任意のウェブ・ページに対する複数の要求が、本来のウェブ・サーバへの単一要求分のコストでまかなうことができる。インターネット・サービス・プロバイダ(ISP)は、一般的に、顧客に対してこのようなキャッシングサービスを提供している。しかしながら、この技術は、次の二つの主な欠点がある。

40

1. グローバルに適用されていないため、キャッシュへのアクセスが可能なウェブ・サイトのユーザさえ、この機能を意図的に起動しなければならない。しかし、多くのユーザは、この機能のことを知らない。

50

2. 動的コンテンツ（遠隔サーバによる即時生成コンテンツ）の場合のように、非常に頻繁に変化するページには適用されない。

【0026】

したがって、動的コンテンツ及び静的コンテンツのキャッシングに対する需要がある。

【0027】

【用語解説／定義】

明示されない限り、すなわち文脈から明らかでなければ、本文で用いる用語、略語、頭字語あるいは科学的なシンボル、そして表記は、本発明が関わる技術的専門分野における通常の意味を持つ。次の用語解説は、本文中の種々の説明に、そして従来技術の引用文に明快さ及び一貫性を与えることを意図している。

10

【0028】

キャッシュ・サーバ

遅延の影響とネットワーク・ボトルネックとを減少させるために、アクセスが頻繁なコンテンツを、そのコンテンツを求めるユーザの近くの戦略的集合ポイントに保存するように最適化したアプリケーション。

【0029】

CARP

キャッシュ・アレイ・ルーティングプロトコル。多数のキャッシュ・サーバを同期させるためのプロトコル。CARPは、ウェブ・オブジェクト・アドレス（URL）を、アレイ内の単一ノードにマッピングする共有ネーム空間を維持する。要求は、そのノードへ経路

20

【0030】

クッキー

インターネット上の「クッキー」の最も一般的な意味は、ウェブ・サーバからウェブ・ブラウザへ送られる一片の情報に言及する。ブラウザ・ソフトウェアは、これを保存し、ブラウザがサーバへ追加の要求を出すときにサーバへ送り返す。用いるクッキーのタイプ、そしてブラウザの設定に応じて、ブラウザは、クッキーを受け入れる、あるいは受け入れない。また、短期間あるいは長期間に渡ってクッキーを保存する。クッキーは、ログインあるいは登録情報、オンライン「買い物カート」情報、ユーザ・プリファレンス等の情報を含んでもよい。サーバは、クッキーを含むブラウザから要求を受信すると、クッキーに保存された情報を用いることができる。例えば、ユーザに送り返すものをカスタマイズする、あるいは特定なユーザの要求をログとしてを保持する。クッキーは、通常、所定の時間後に失効するように設定されており、ブラウザ・ソフトウェアが閉じられるまでメモリに保存され、ブラウザが終了するとき、「失効時」に達していない場合、ディスクに保存される。

30

【0031】

動的コンテンツ

規則的に更新される「ライブ」コンテンツ。動的コンテンツの例としては、気象ウェブ・サイト上の「現在の温度」表示、検索結果、あるいはニュース・ウェブ・サイト上の「現在のトップ・ヘッドライン」項目がある。

40

【0032】

HTTPサーバ

クライアント・エージェント（ブラウザ）へウェブ・ページを提供できるように、HTTPプロトコルを実行するサーバ。HTTPサーバは、ウェブ・ページが外部のプログラムを呼び出すためのインタフェースをサポートする。また、情報及び認証を安全に交換するための暗号化メカニズムと、コンテンツへのアクセスを制御するためのアクセス制御メカニズムとをサポートする。

【0033】

ICP

インターネット・キャッシュ・プロトコル。多数のキャッシュ・サーバを同期させるため

50

のプロトコル。キャッシュ・サーバは、ミスをする度に、すべてのピア・ノードにメッセージを送って、各ノードがコンテンツを持っているかどうかを尋ねる。要求サーバは、コンテンツの要求を出し、それからユーザへ転送する。

【0034】

プロキシ・サーバ

プロキシ・サーバは、ユーザとインターネットとの間に仲介者として機能するため、企業は、セキュリティを保証し、そして管理上のコントロールができ、また、キャッシングサービスをも提供することができる。プロキシ・サーバは、通常、企業ネットワークを外部のネットワークから分離するゲートウェイ・サーバあるいはその一部に、または企業ネットワークを外部からの侵入から保護するファイアウォールに関連づけられる。

10

【0035】

ルータ

ルータは、ポイントからポイントのリンクのコレクションから完全に相互接続されたネットワークを構築するデバイスである。インターネット上のルータは、ネットワークの局所的区分、特にネットワークの構成上、ローカルシステムにどれくらい近いかに関する情報を交換する。インターネット内において、ポイントから他のポイントへ到達するためのマップを集散的に作成する。パケットは、交換したマッピング情報に基づいて経路が決められ、最後のルータが目標システムに直接結合する。

【0036】

静的コンテンツ

変更のない、あるいは滅多に変更がないファイルとして保存された、ウェブページの「固定」または長期間不変な成分。

20

【0037】

スイッチ

典型的にインターネットの周囲に存在する高速ネットワーク・デバイス。スイッチは、より低価格でより高い性能を提供することにおいてルータとは異なるが、機能が限られている。通常のスイッチは、局所的に通信の経路を決めることができるが、高速インターネット・バックボーン内に見られるルーティングの複雑さには関与しない。スイッチは、しばしば、キャッシュ可能な通信をキャッシングシステムへ向けるために用いられるため、キャッシングにおいて重要な役割を持つ。

30

【0038】

HTML

ハイパーテキスト・マークアップ言語。標準方法でユーザ・エージェント（ブラウザ）内に表示が可能なように、テキストにタグを付けるための標準汎用マークアップ言語（SGML）に基づく仕様。

【0039】

HTTP

ハイパーテキスト送信プロトコル。ワールド・ワイド・ウェブの基礎であるTCP/IP上で稼働するアプリケーション・レベルのプロトコル。

【0040】

IP

インターネット・プロトコル。TCP/IPプロトコル・スイートのためのネットワーク層であり、無接続の最も効率的なパケット交換プロトコルである。

40

【0041】

IPアドレス

通常10進数表記法で表すインターネット・プロトコルによって定義される32ビットのアドレスであって、インターネット上の各コンピュータをユニークに識別する。

【0042】

プロトコル

コンピュータが情報を交換するための、合意上の技術的なルールのセット。

50

【0043】

URL

ユニフォーム・リソース・ロケータ。インターネット・サイトのアドレスを定める方法であって、アクセス・プロトコルと、IPアドレスあるいはDNSネームとを含む。例えば、「http://www.domain.com」。

【0044】

ユースネット

ユーザのネットワークに対する短縮形。インターネット上に存在する何万という掲示板のコレクションであって、各々が、種々のトピック専用のニュース・グループと呼ばれるディスカッション・グループを含む。メッセージを公表し、ネットワーク・ニュース転送プロトコル(NNTP)を介して返答を受ける。

10

【0045】

ウェブ・サーバ

HTTPサーバを参照。

【0046】

【本発明の簡単な説明】

本発明の目的は、WWWサーバのバンド幅の利用を減少させて、WWWサイトのQoSを改善するための技術を提供することである。

【0047】

本発明によれば、インターネット上の動的コンテンツを持つオブジェクトをキャッシュするための技術は、概して、動的コンテンツのコピーを保存して更新するためにインターネット内にキャッシュを配置することからなる。主要なインターネット・スイッチングロケーション、ダイヤル・イン集合ポイント、そして企業ゲートウェイからなるグループから選択した箇所にキャッシュを配置する。また、静的コンテンツのコピーをもキャッシュに保存して更新してもよい。

20

【0048】

本発明の特徴によれば、オブジェクトの更新特性が判定され、更新特性に基づいて、オブジェクトに対する残存時間(TTL)パラメータが調整される。一般的に、オブジェクトは、そのTTLが、その経過時間より少ない場合に更新される。オブジェクトに対するTTLを、次のように調整してもよい。

30

(i) 誤り確率を所定の誤り確率閾値よりも少なく維持する、

(ii) 誤り率を所定の誤り確率閾値よりも少なく維持する、あるいは

(iii) 遅延時間を所定の遅延閾値よりも少なく維持する。

【0049】

本発明によれば、サーバから生じる、動的コンテンツを持つオブジェクトに対するユーザ要求に応答するための方法は、オブジェクトのコピーをキャッシュに保存し、オブジェクトに対する残存時間(TTL)を設定し、キャッシュにおいてユーザ要求を受け、TTLが経過時間より多い場合はオブジェクトの保存コピーでユーザ要求を満たし、そして、保存コピーのTTLが経過時間より少ない場合はオブジェクトの更新コピーをフェッチして更新コピーでユーザ要求に応答する。

40

【0050】

本発明の特徴によれば、オブジェクトに対するTTLは、最初、合理的な下限(Tmin)に設定され、その後、オブジェクトが実際に変化する頻度に基づいて調整される。

【0051】

本発明の特徴によれば、キャッシュがサーバからオブジェクトをフェッチする度に、キャッシュは、次の処理を行う。

a. 同じオブジェクトに対するもう一つのフェッチが進行中である場合、以前のフェッチが完了するのを待つ。

b. サーバからオブジェクトをフェッチする。

c. キャッシュされているコピーを、もし存在するなら、フェッチしたオブジェクトで置

50

き換えるが、この場合は、最後にフェッチして以来オブジェクトが変化したかどうかを判定するために比較した後に置き換える。

d. オブジェクトの変化統計値を初期化する、あるいは相応にアップデートする。

e. サーバの応答に応じて、静的あるいは動的コンテンツと、オブジェクトに標識する。そして、

f. オブジェクトが動的である場合は、オブジェクトの変化統計値から判定した変化 (τ) 間の平均時間、オブジェクト・アクセス統計値から判定した単位時間ユーザ要求数 (η)、そして次の処理 (A-E) の一つから TTL (T) を適当な値に設定する。

A. 最大誤り確率。

B. 最大誤り率。

C. 最大遅延。

D. T が最低になる上記 (A、B、C) の組合せ。または、

E. T_{min} と T_{max} とが挟む「妥当な」値の所定範囲内に T を留める上記 (A、B、C) の組合せ。

【0052】

キャッシュは、専用マシンであって、ウェブ・ブラウジングが、本来のサイトへの全行程を通過する代わりにキャッシュを通過するよう、多くの異なるロケーションに、理想的には、そのロケーション内の最大数のユーザインターネット接続を提供する ISP のネットワーク内に配置することが好ましい。このようにすれば、

—それらの ISP、そして程度は小さいが、隣接する ISP のユーザは、通信の大部分が ISP の内部ネットワークの範囲に、あるいは近くに留まり、負荷の高いインターネット・バックボーンを通る必要がないため、莫大な QoS 及びスピード改善を得ることになる。また、

—キャッシュが負荷の大部分を吸収するため、本来のウェブ・サイトは、もはや、それほどビット・レートが必要としない。

【0053】

ウェブ・サイトの多くが、全内容が動的なコンテンツを提供する傾向にあるため、動的コンテンツも静的コンテンツと同様にキャッシュさせる。しかしながら、コンテンツには変わる時期を示す指標がないので、キャッシュの問題は単にコンテンツを覚えておくということだけではない。単にキャッシュ・コピーを提供する代わりに、周期的に再ロードする必要がある。

【0054】

したがって、キャッシュは、対応するキャッシュ・コピーが任意の時間 (残存時間、「TTL」) リフレッシュされていないときにはいつでもページを再ロードする (サーバからオブジェクトをフェッチする)。TTL はページの「有効期間」であると考えることができる。キャッシュは、まず、動的オブジェクトに対する TTL を妥当な下限、 T_{min} に設定する。それから、時間の経過とともにキャッシュが数回ページを再ロードしなすとき、ページのコンテンツが実際に変わる頻度を探知し、そのページに対する TTL を相応に調整する。もしページの本来のサーバが TTL を指定するなら、この「TTL」技術は、静的なページに対する共通のキャッシング方法をまねる。動的ページのサーバは TTL を指定しないので、キャッシュは単独で妥当な TTL を設定しなければならない。

【0055】

したがって、

—動的コンテンツを含むウェブ・サイト全体をキャッシュすることが可能である。

—キャッシュと本来のサイトとの間の通信をより良く管理することができる。

—キャッシュがページを再ロードする前にページが変化するケースでは、数人のユーザが古いコンテンツを受ける可能性があるが、(ネットワークが飽和している場合にはキャッシュが無くとも起きる可能性がある) 遅延及び誤り確率を制限することは可能である。

—システムは、各ページへの要求数、そしてページの実際の更新頻度に応じて、リアルタイムで適応することができる。

10

20

30

40

50

【0056】

本発明の他の目的、特徴及び利点は、次の説明文から明らかになる。

【0057】

添付図面に示す例を参照にして、本発明の好適実施例を詳細に説明する。次の図は、限定せずに説明することを意図している。本発明は、概して、これらの好適実施例に関連して説明するが、本発明の範囲を、これらの特定な実施例に制限することを意図していない。

【0058】

ここに示すフローチャート内の長方形の囲みは、通常、実行される連続したステップを、概して端部が尖った長方形の囲みは、二つの相互に排他的な結果を持つ決定ステップ（テスト）を表す（「Y」＝イエス、「N」＝ノー）。そして空円は、ステップあるいはテストではなく、フローチャート内の二つ以上の経路が収束する接合点を示す。 10

【0059】

本発明の好適実施例の構造、作用及び利点は、添付図を参照しながら次の説明文を考慮すれば、より明らかになる。

【0060】

〔発明の詳細な説明〕

図1は、インターネット環境をかなり単純化した概要図で、本発明のキャッシングシステムの実施例を示す。概して、ユーザ102は、サーバ（例えば、ISP）104によってインターネット上に利用可能な「オブジェクト」（例えば、ウェブ・ページ）に対する「要求」を行う（オブジェクトは、典型的に、コンテンツ提供者から生じる、図示せず）。 20
スイッチ106は、インターネットへキャッシュ（キャッシュ・サーバ）108を繋ぐインタフェースとして機能する。キャッシュはウェブ・ページのコピーを含んでもよい。ユーザ要求に対して二つの「応答」が可能である。サーバが要求を「処理する」（すなわち「サービスをする」）、あるいはキャッシュで要求を「満たす」。後者の場合は、第一に、コンテンツがキャッシュへ「転送され」なくてはならない。その後、サーバから周期的に更新ウェブ・ページ・コンテンツを「フェッチ」（すなわち「再ロード」）してもよい。次のディスカッションでも、可能な限りこれらの用語を用いる。

【0061】

ゲールド氏の記事（例えば、ページ18、図4）で論じられているように、スイッチ106は、ネットワーク通信を観察し、IPレベルを超えるプロトコルに基づいてルーティング決定を行うことが可能な高性能プロセッサでもよい。その結果、スイッチは、HTTP（及び他の）通信をキャッシュ（108）へ導き、残りの通信を直接インターネットへ送信する。これは、複数の可能なシステムアーキテクチャの一つの典型である。どのアーキテクチャを用いるかは、キャッシュの実行位置、キャッシュの主要な目的、そして通信の性質を含む複数の要因に依存する。 30

【0062】

ゲールド氏の記事では、さらに次のように論じられている。

【0063】

非透過キャッシュは明確に可視であり、そのキャッシュを用いるブラウザあるいは他のキャッシュは、通信をそのキャッシュへ導くように明確に構成される。この場合、キャッシュは、ブラウザのプロキシ・エージェントとして機能し、可能なときは要求を満たし、必要なときは発信サーバへ要求を転送する。非透過キャッシュは、しばしば、ゲートウェイあるいはファイアウォールの一部として機能し、そして多くの異なるアプリケーションを扱う、より大きなプロキシ・サーバの構成要素である。（ゲールド、ページ16－17） 40

【0064】

透過キャッシュは、ネットワーク経路内に置かれ、ブラウザには見えないように機能する。ISP及び企業バックボーン稼働に対しては、透過構成が好まれる。これは、ブラウザがキャッシュを見いだすように構成されると、ユーザ・サポートの全体的な管理及びサポートの負担が最小になるためである。（ゲールド、ページ17） 50

【0065】

キャッシュは、キャッシングの利点を最大にするため、透過的に具現すべきである。非透過の実施例では、ブラウザを手作業で設定して、コンテンツの要求をキャッシュ・サーバへ導くようにする必要がある。透過構成では、ブラウザを再設定せずに、キャッシュの利点をクライアントに提供することができる。ユーザは自動的にキャッシングの利益を得る。(ゲールド、ページ17)

【0066】

企業の環境では、自動的なブラウザ設定を介して、あるいはネットワーク上の通信を途中で捉えることによって、透過性を得ることができる。ネットスケープ及びマイクロソフトの両方は、ブラウザの大きなネットワークを中央で管理し、ユーザのPCにインストールされたブラウザの設定をカスタマイズするユーティリティを提供している。ブラウザのプラグインも自動設定を提供する。このアプローチはユーザに対して透過的であるが、継続的な管理努力を必要とする。(ゲールド、ページ17)

【0067】

透過キャッシングを提供する複数の選択肢がある。

一キャッシュをルータであるかのように構成し、インターネットに基づくすべての通信をキャッシュへ向けることができる。これは、ブラウザの設定を必要としない透過構成であり、ブラウザあるいは下流のキャッシュは、キャッシュの存在に気づかないが、それでも利益を得る。欠点は、キャッシュが位置するシステムが、そのリソースのいくらかをルーティングに割かなければならないことであり、キャッシュはネットワークの機能上不可欠な部分になる。方針に基づくルーティングを行う精巧なルータ設定は、単にHTTP(TCPポート80)通信をキャッシュへ導き、失敗した場合にはキャッシュを迂回して、通信を直接インターネットへ送ることによって、これらの問題のいくつかを最小限にする。(ゲールド、ページ17)

一好評なオプションとしては、インターネットへキャッシュをインタフェース接続するレイヤ4スイッチを用いるやり方がある(図4参照)。現在、アルテオン(Alteon)、ファウンドリ(Foundry)、アローポイント(ArrowPoint)、その他によって提供されているこれらのスイッチは、ネットワーク通信を調べて、IPレベルを超えるプロトコルに基づいてルーティング決定を行うことが可能な高性能プロセッサである。その結果、スイッチは、HTTP(そして他の)通信をキャッシュへ導き、残りの通信を直接インターネットへ送信する。スイッチはHTTPの要求を解析して、そして求められたURLに基づいてキャッシュ農場で特定のノードに要求を送ることができる。インテリジェント・スイッチを用いることによって、不必要なネットワーク通信をキャッシュから切り離し、有用性に対するデザインを単純にし、特定のURLに基づくキャッシュ・ファーム上のロードを分配することができる。(ゲールド、ページ18)(これに類似するアーキテクチャが、図1に関して上記に説明されている。)

一透過性に対するもう一つのオプションとしては、ウェブ・キャッシュ・コントロール・プロトコル(WCCP)である。WCCPは、シスコ・システムズ(Cisco Systems)によって開発されたもので、シスコISOに基づくルータを用いるネットワーク内にウェブ・キャッシュを透過的にインストールすることを可能にする。WCCPを用いれば、HTTP通信が、発信サーバの代わりにウェブ・キャッシュへ向け直すことができる。WCCPは、ネットワーク・アーキテクチャに対する変更を必要とせず、透過ウェブ・キャッシングの実施を単純化する。(ゲールド、ページ18)

一ウェブ・プロキシ自動検出プロトコル(WPAD)は、新しく提案された標準プロトコルであり、ブラウザ、ストリーミングメディア・クライアント、そして他のインターネット・クライアント・ソフトウェアと統合することによって、末端ユーザの設定を必要とせずに、ネットワーク上のキャッシュ及びサービスを自動的に見つけるようにデザインすることができる。WPADは、ルーティングあるいはスイッチング装置を用いる既存のキャッシュ透過性による解決策に対する、柔軟な、そしてベンダーに中立なソフトウェアの選択肢を提供する。将来的に、WPADが使用可能なクライアント・ソフトウェアは、領域

内のユーザを埋め込みのネットワーク・サービスへ自動的に接続することによって、これらのサービスを提供するネットワーク提供者とユーザとの両方に単純さを提供する。(グールド、ページ18-19)

【0068】

キャッシングシステムを用いることによって、ウェブ・サーバ・サイトの性能を最適化し、ウェブ・ブラウザによるユーザのインターネット・アクセスを高速にすることができる。逆方向のプロキシ構成においては、キャッシングシステムが、一つ以上のウェブ・サーバの前に位置し、それらのサーバへの通信を途中で捕らえてスタンドインになる、すなわち一つ以上のサーバの代用となる。ネットワーク内にキャッシュ・サーバを配置し、主催するコンテンツに対する分散ネットワークを構築することができる。必要ならば、プロキシ・キャッシュ・サーバは、発信サーバから、動的な、そして他の短命なコンテンツを求め、これは、サイトからのコンテンツを、発信サーバの代わりにローカル・キャッシュから提供することを可能にする。プロキシ・サーバは、高性能で効率的な処理を行うように最適化させることができる。そしてリソースを節約し、発信サーバを静的コンテンツの提供から解放することができる。これらの利点は、負荷をバランスさせる機能を含み、有用性を維持するピーク需要に対する保証を提供し、また有用性を高度に維持するために、コンテンツの動的なコピーを提供する。(グールド、ページ20)

【0069】

ネットワーク上におけるキャッシュの最良な配置を決める三つの一般的な特徴がある。
ーネットワーク通信内の渋滞ポイント。ネットワーク通信の大多数が通過する箇所、そのためにキャッシュ・サーバに可視であるロケーションがある。このような箇所に配置すれば、容易に回避されてしまうような箇所に位置する場合に比べ、キャッシュでより多くの要求を処理し、より多くのコンテンツを保存することができる。

ー高ネットワーク負荷ポイント。通信が多い状態は、より高いキャッシュ使用率を可能にするため、利点が多い。

ーキャッシュに対して最大の経済的利益を生じるロケーション。高いキャッシュ・ヒット率からユーザが利益を得ると共に、また上流のビット・レート要求を減少させるポイントは、アクセス・プロバイダに、QoSにおける利点、そして経済性の両方を提供する。これらの特徴を持つロケーションは、典型的に、主要なインターネット・スイッチング・ロケーション、ダイヤル・イン集合ポイント、あるいは企業ゲートウェイに見られる。(グールド、ページ20)

【0070】

これらの特徴を持つロケーションは、典型的に、次のものを含む主要なインターネット・スイッチング・ロケーション、ダイヤル・イン集合ポイント、あるいは企業ゲートウェイに見られる。

POP及びダイヤルアップ(グールド、ページ21、図7を参照)

NAPS(グールド、ページ22、図8を参照)

衛星(グールド、ページ22参照)

国際ゲートウェイ(グールド、ページ23、図9を参照)

ウェブ・ホスティング(グールド、ページ24、図10を参照)

ラストマイル(グールド、ページ25、図11を参照)

企業ゲートウェイ(グールド、ページ25、図12を参照)

【0071】

本技術における通常の技能を持つ者は、上記の説明から、本発明のキャッシュを配置すべき位置を容易に決めることができる。一般的に、本発明のキャッシュは、静的コンテンツを提供するキャッシュがある箇所に配置する、あるいは、静的コンテンツに対する要求を満たす既存のキャッシュに組み込み、下記の技術によって可能な追加の機能を加えることができる。または、分離した専用マシン(コンピュータ)として提供してもよい。

【0072】

図2は、静的及び/あるいは動的コンテンツに対するユーザの要求をどのようにキャッシ

10

20

30

40

50

ユによって処理するかを示すフローチャートである。

【0073】

第一のステップ202では、オブジェクトに対するユーザの要求に対して、キャッシュは、要求されたオブジェクトがキャッシュ内にあるかどうかを判定する。無い(N)場合は、ユーザ要求は、要求を満たすためのサーバへ送られる。同時に、ステップ204において、キャッシュは、オブジェクトに対する同じ、あるいはもう一人のユーザからの次の要求を予期して、サーバからオブジェクトをフェッチする。

【0074】

概して、この例では、オブジェクトに対するすべての要求は、ユーザに対して透過なキャッシュ・サーバを通過すると仮定している。情報要求を途中で捉え、応答を、キャッシュしたローカルコピーから提供するのか、あるいは遠隔の情報源から提供するのかを決定する。ローカルソースから情報をフェッチした後、キャッシュ・サーバは、それを局所的に保存すべきか、もしそうであるなら、どれぐらいの期間保存するのかを決める。ローカルコピーから提供することができる情報の要求は、「キャッシュ・ヒット」として知られる。逆に、ローカルに保存されていない情報の要求は、「キャッシュ・ミス」として既知である。保存判定アルゴリズムが良くデザインされていれば、キャッシュ・ヒットの確率はかなり改善し、(QoSに関する)ユーザ要求への外見上の応答時間は減少する。さらに、ローカルにキャッシュしたコンテンツによって満たされるすべての情報要求(キャッシュ・ヒット)は、外部ネットワーク上の通信を減少させるため、外部ネットワーク上の応答時間も短くなる。

【0075】

要求オブジェクトがキャッシュ内にある場合、次にステップ206で、要求オブジェクトが静的であると標識されているかどうかを判定する。そうである(Y)場合は、ステップ208において、オブジェクトの「経過時間」(最後にリフレッシュされたときから経過した時間)を、(そのページの本来のサーバがTTLを指定している場合、サーバに最近の変更時間等を尋ねて)TTLに比較する等の、静的オブジェクトをキャッシュするための適当な標準アルゴリズムを用いて、キャッシュ・コピーを更新すべきか、あるいはユーザ要求を満たすためにそれを用いるべきかどうかを判定する。

【0076】

要求オブジェクトがキャッシュ内にあり、動的である(N、ステップ206)場合は、キャッシュ・コピーのTTL(「有効期間」)が、その経過時間より少ないかどうかをステップ210で判定する。

【0077】

キャッシュ・コピーのTTLが経過時間よりも少ない(より小さい数)である場合(Y、ステップ210)は、「新鮮でない」と考え、ステップ212において、キャッシュは、一オブジェクトのアクセス統計値(最後の数分、最後の時間等の単位時間あたりのユーザ要求数)を更新し、そして一本来のサーバからオブジェクトをフェッチする。

【0078】

キャッシュ・コピーのTTLがその経過時間に等しい、あるいは多い場合(N、ステップ210)、「新鮮である」と考え、ステップ214において、キャッシュは、一キャッシュ・コピーを用いて要求を満たし、そして一オブジェクトのアクセス統計値を更新する。

【0079】

オプションとして、キャッシュ・コピーの経過時間とTTLとの間の時間差が任意の時間よりも少なく、かつ最近のユーザ要求数が任意の率よりも多い場合は、「古いもの」ではあるが「人気が高い」と考え、キャッシュはサーバからオブジェクトをフェッチするが、これは「予測リフレッシュ」と呼ばれる。

【0080】

さらに、キャッシュがサーバからオブジェクトをフェッチする(ステップ204及び21

10

20

30

40

50

2を参照) 度に、ステップ216において、キャッシュは次の処理を行う。

a. (例えば、以前にユーザ要求があったために) 同じオブジェクトに対するもう一つのフェッチが進行中である場合、キャッシュは、その要求を繰り返さず、先行するフェッチが完了するのを待つ。

b. サーバからオブジェクトをフェッチする。

c. もしそのキャッシュ・コピーが存在するなら、最後にフェッチしてからオブジェクトが変更されているかどうかを判定するために比較した後、最新のオブジェクトによって置き換える。

d. それ相応に、オブジェクトの変化統計値(単位時間あたりの変化数)を初期化する、あるいは更新する。

10

e. 本来のサーバからの応答(変更時間の、あるいはゼロ以外のTTL等の存在)に応じて、オブジェクトを静的あるいは動的コンテンツと標識する。

f. もしオブジェクトが動的であるなら、次の事項に応じてTTL(T)を適当な値に設定する。(オブジェクトの変化統計値から決定した)変化間の平均時間 τ 、(オブジェクト・アクセス統計値から決定した)単位時間当たりのユーザ要求数 η 、そして次の処理(A-E)から選択した一つ。

【0081】

A. 最大誤り確率(p_0)。これは、任意の時間Wを超える時間において、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした要求数の、全要求数に対する平均比率である。

20

$$T = (W + \tau p_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right\}$$

【0082】

B. 最大誤り率(n_0)。これは、任意の時間Wを超える時間において、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした単位時間当たりの平均要求数である。

30

$$T = \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

【0083】

C. 最大遅延(D_0)。これは、オブジェクト変化と、キャッシュ・コピーのリフレッシュとの間の平均時間である。

$$T = \frac{6D_0}{1 + 3 \frac{D_0}{\tau}}$$

40

【0084】

D. Tが最も低くなる上記(A、B、C)の組合せ。あるいは、

【0085】

E. Tを T_{min} と T_{max} とが挟む「妥当な」値の所定の範囲内に留める上記(A、B、C)の組合せ。

【0086】

上記方程式には、次のパラメータを用いている。

Tは、動的オブジェクトに対する残存時間である。

Wは、本来のオブジェクトが変化してから(すなわち、古さを表す)任意の時間である

50

τ は、オブジェクトの変化統計値からの判定される、変化間の平均時間である。

n は、単位時間当たりのユーザ要求数（例えば、頻度）である。

p_0 は最大誤り確率である。これは、任意の時間 W を超える時間において、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした要求数の、全要求数に対する平均比率である。

η_0 は最大誤り率である。これは、任意の時間 W を超える時間において、対応する本来のオブジェクトが変化したキャッシュ・コピーを用いて満たした単位時間当たりの平均要求数である。

D_0 は最大遅延である。これは、オブジェクト変化と、キャッシュ・コピーのリフレッシュとの間の平均時間である。 10

【0087】

【TTLの選択及び調節】

オブジェクトの変化間の平均時間（ τ ）だけで、他のパラメータをとらない単純なモデルは、指数減少モデル、すなわち、時間 t の間にオブジェクトが変化しない確率を式 $e^{-t/\tau}$ であると考え、 $t=0$ に対しては 1 であるが、 t が $+\infty$ になるにつれ 0 に近づく。

【0088】

キャッシュ・コピーのリフレッシュ率に比較しながら、多くのユーザ要求がある状態を考察すると、キャッシュは、ユーザによって作動されるが、規則的な間隔で更新する。 20

【0089】

変化の平均頻度

さて、モデルの一貫性、特に、任意のオブジェクトの平均更新頻度が $1/\tau$ であるかどうかをチェックする。 $p_k^c(t)$ を、時間 t に渡って k が変化する確率であるとする。構成上、

$$p_0^c(t) = e^{-t/\tau}$$

そして、 $k > 0$ に対して、

$$p_k^c = \int_0^t p_{k-1}^c(u)(1 - e^{-du/\tau})e^{-(t-u)/\tau} du = \int_0^t p_{k-1}^c(u) \frac{du}{\tau} e^{-(t-u)/\tau} \quad 30$$

（ $[0:t]$ に渡って k 変化が生じる確率は、 0 と u との間に起こった $k-1$ 変化の確率の u に渡る和である。というのは、 u と $u+du$ との間で変化があったが、 $u+du$ と t との間には変化がなかった。）

【0090】

繰り返しから次のように演繹する。

$$p_k^c(t) = \frac{1}{k!} \left(\frac{t}{\tau} \right)^k e^{-t/\tau}$$

$$\begin{aligned} \int_0^t \frac{1}{k!} \left(\frac{u}{\tau} \right)^k e^{-u/\tau} \frac{du}{\tau} e^{-(t-u)/\tau} &= \frac{1}{k! \tau^{k+1}} e^{-t/\tau} \int_0^t u^k du \\ &= \frac{1}{k! \tau^{k+1}} e^{-t/\tau} \frac{t^{k+1}}{k+1} \end{aligned} \quad 40$$

$$= \frac{1}{(k+1)!} \left(\frac{t}{\tau} \right)^{k+1} e^{-t/\tau}$$

【0091】

さて、 t 時間に渡る平均変化数は（平均の定義から）次のようになる。

$$\begin{aligned}
 \sum_{k=0}^{+\infty} k \times p_k^c(t) &= \sum_{k=1}^{+\infty} k \times p_k^c(t) \\
 &= \sum_{k=1}^{+\infty} k \times \frac{1}{k!} \left(\frac{t}{\tau}\right)^k e^{-t/\tau} \\
 &= e^{-t/\tau} \times \sum_{k=1}^{+\infty} \frac{1}{(k-1)!} \left(\frac{t}{\tau}\right)^{k-1} \\
 &= e^{-t/\tau} \left(\frac{t}{\tau}\right) \times \sum_{k=0}^{+\infty} \frac{1}{k!} \left(\frac{t}{\tau}\right)^k \\
 &= e^{-t/\tau} \left(\frac{t}{\tau}\right) e^{t/\tau} \\
 &= \frac{t}{\tau}
 \end{aligned}$$

10

【0092】

したがって、変化は $1/\tau$ の平均頻度を持つことが分かる。

【0093】

平均誤り確率

誤り確率は、本来のサーバ上のオブジェクトが変化し、キャッシュ・コピーが任意の時間 W を超えるだけ古いにも拘わらず、ユーザがキャッシュ・コピーを受信する場合のパーセンテージであると定義する。図3は、（水平軸が秒を示す）時間に対する（垂直軸がパーセンテージを示す）パーセンテージ・エラーを示す。例えば、 $W=15$ s、 $\tau=60$ s、 $T=30$ s。グラフは、 W 秒を超える時間に対するキャッシュ・コピーが「新鮮でない」確率を示す。グラフ中、 W は15秒であるため、1秒から15秒の間の確率はゼロである。それから、確率は、ぎざぎざのパターン302で示す再フェッチが起きるとき（ $T=30$ s）、すなわち30秒まで増大する。このパターンは30秒ごとに繰り返す。30秒から45秒の間は、確率は再びゼロであり、再フェッチが45秒で起こる。それから、ぎざぎざのパターン304で示すように、確率は60秒まで増大する。75秒から90秒の間にも、ぎざぎざのパターン306で同様の結果を示す。

20

30

【0094】

キャッシュが一定間隔でオブジェクトのコピーを更新するようにサポートするとすれば、時間 $[0; T]$ に渡る計算を行うことができる。

【0095】

コンテンツが古い期間内の平均時間 E は、0 から u の間には変化がなかったが、 u から $u + du$ の間に変化があった確率の、すべての期間 $[u; u + du]$ に渡る和に、 W を超えるだけ古いままコンテンツが存在する時間を掛けたものである。すなわち、

$$E = \int_0^{T-W} e^{-u/\tau} \frac{du}{\tau} (T-W-u)$$

$$= (T-W)(1 - e^{-\frac{T-W}{\tau}}) - \tau \int_0^{T-W} \frac{u}{\tau} e^{-u/\tau} \frac{du}{\tau}$$

$$= (T-W)(1 - e^{-\frac{T-W}{\tau}}) - \tau \left(-\frac{T-W}{\tau} e^{-\frac{T-W}{\tau}} + 1 - e^{-\frac{T-W}{\tau}} \right)$$

10

$$E = (T-W) - \tau(1 - e^{-\frac{T-W}{\tau}})$$

あるいは、

$$\alpha = \frac{T-W}{\tau}$$

$$E = \tau(\alpha - 1 + e^{-\alpha})$$

20

である。

【0096】

これから、誤り確率を次のように演繹することができる。

$$P_{err} = \frac{E}{T} = \frac{\tau}{T}(\alpha - 1 + e^{-\alpha})$$

【0097】

処理が容易な範囲を計算することができる。

$$F(\alpha) = \frac{\alpha^3}{6} - \frac{\alpha^2}{2} + \alpha - 1 + e^{-\alpha}$$

30

【0098】

その導関数は次のようになる。

$$F'(\alpha) = \frac{\alpha^2}{2} - \alpha + 1 - e^{-\alpha}$$

$$F''(\alpha) = \alpha - 1 + e^{-\alpha}$$

$$F'''(\alpha) = 1 - e^{-\alpha}$$

40

【0099】

$F(0) = F'(0) = F''(0) = 0$ 、そして $\alpha \geq 0$ に対して $F'''(\alpha)$ がプラスであるため、 $F''(\alpha)$ 、 $F'(\alpha)$ 、そして究極的に $F(\alpha)$ も同様である。このため、

$$\frac{\alpha^2}{2} - \frac{\alpha^3}{6} \leq \alpha - 1 + e^{-\alpha} \leq \frac{\alpha^2}{2}$$

【0100】

$\alpha \geq 1$ のケースに対する、より良い、見つけやすい範囲があるが、これは興味深いケー

50

ス ($T - W \geq \tau$) ではない。したがって、

$$\tau \left(\frac{\alpha^2}{2} - \frac{\alpha^3}{6} \right) \leq E \leq \tau \frac{\alpha^2}{2}$$

$$\frac{\tau}{T} \left(\frac{\alpha^2}{2} - \frac{\alpha^3}{6} \right) \leq p_{err} \leq \frac{\tau}{T} \frac{\alpha^2}{2}$$

【0101】

任意の誤り確率を保証する最大 T T L

上限は、誤り確率が任意の閾値 p_0 よりも少ないことを保証できる T T L に設定すべきである。

10

$$P_{err} \leq p_0 \Leftrightarrow \frac{\tau}{T} (\alpha - 1 + e^{-\alpha}) \leq p_0$$

$$\Leftrightarrow \frac{\tau}{T} \frac{\alpha^2}{2} \leq p_0$$

$$\Leftrightarrow \frac{(T - W)^2}{\tau} \leq 2p_0 T$$

$$\Leftrightarrow T^2 - 2(W + \tau p_0)T + W^2 \leq 0$$

20

$$\Leftrightarrow W + \tau p_0 - \sqrt{(W + \tau p_0)^2 - W^2} \leq T \leq W + \tau p_0 + \sqrt{(W + \tau p_0)^2 - W^2}$$

$$\Leftrightarrow W \leq T \leq (W + \tau p_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right\}$$

【0102】

したがって、許容可能な T T L は、次のようになる。

$$T = (W + \tau p_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right\}$$

30

【0103】

任意の誤り率を保証する最大 T T L

これは、 p_0 を n_0 / n で置き換えただけで、誤り確率に対するものと同じである。

$$n_{err} = np_{err}$$

$$n_{err} \leq n_0 \Leftrightarrow p_{err} \leq \frac{n_0}{n}$$

$$\Leftrightarrow T \leq \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

40

【0104】

したがって、許容可能な T T L は、次のようになる。

$$T = \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

【0105】

任意の TTL に対する平均遅延

キャッシュ・コピーと本来のオブジェクトとの間の、後者が任意の期間中に変化した場合の、平均遅延 D を考慮することは有用である。まず、0 から T の間の時点 u に変化が起こったときの平均遅延である $\langle \Delta t(u) \rangle$ を計算する。 10

$$\begin{aligned} \langle \Delta t(u) \rangle &= \frac{1}{T} \int_0^T dv(v-u) \\ &= \frac{(T-u)^2}{2T} \end{aligned}$$

【0106】

これから、 $[0; T]$ に渡って u と $u+du$ との間に変化が起こった確率と、 $\langle \Delta t(u) \rangle$ とを掛けたものを積分することによって、全体的な遅延 $\langle \Delta t \rangle$ を演繹することができる。 20

$$\begin{aligned} \langle \Delta t \rangle &= \int_0^T e^{-v/\tau} \frac{du}{\tau} \frac{(T-u)^2}{2T} \\ &= \frac{1}{2T} e^{-T/\tau} \int_0^T e^{\frac{T-u}{\tau}} \frac{du}{\tau} (u-T)^2 \\ &= \frac{\tau^2}{2T} e^{-T/\tau} \int_{\frac{T}{\tau}}^0 x^2 e^{-x} dx \end{aligned}$$

$$\begin{aligned} &= \frac{\tau^2}{2T} e^{-T/\tau} \left\{ \left(\frac{T}{\tau} \right)^2 e^{T/\tau} + \int_{\frac{T}{\tau}}^0 2xe^{-x} dx \right\} \\ &= \frac{\tau^2}{2T} e^{-T/\tau} \left\{ \left(\frac{T}{\tau} \right)^2 e^{T/\tau} - 2 \frac{T}{\tau} e^{T/\tau} + 2 \int_{\frac{T}{\tau}}^0 e^{-x} dx \right\} \\ &= \frac{\tau^2}{2T} e^{-T/\tau} \left\{ \left(\frac{T}{\tau} \right)^2 - 2 \frac{T}{\tau} e^{T/\tau} (1 - e^{-T/\tau}) \right\} \\ &= \frac{\tau^2}{2T} \left\{ \left(\frac{T}{\tau} \right)^2 - 2 \frac{T}{\tau} - 2(e^{-T/\tau} - 1) \right\} \\ \langle \Delta t \rangle &= \frac{\tau^2}{T} \left\{ \frac{1}{2} \left(\frac{T}{\tau} \right)^2 - \frac{T}{\tau} + 1 - e^{-T/\tau} \right\} \end{aligned}$$

【0107】

$\langle \Delta t \rangle$ の計算は、 $[0; T]$ 内において変化があったことを考慮していないため、対応する確率によって除して、 D を得る。 40

$$D = \frac{\langle \Delta t \rangle}{1 - e^{-T/\tau}}$$

$$= \frac{1}{1 - e^{-T/\tau}} \frac{\tau^2}{T} \left\{ \frac{1}{2} \left(\frac{T}{\tau} \right)^2 - \frac{T}{\tau} + 1 - e^{-T/\tau} \right\}$$

そして、

$$\frac{\alpha^3}{6} - \frac{\alpha^2}{2} + \alpha - 1 + e^{-\alpha} \geq 0$$

10

そして、

$$\frac{\alpha^2}{2} - \alpha + 1 - e^{-\alpha} \geq 0$$

であるため、

$$D \leq \frac{1}{\frac{T}{\tau} - \frac{1}{2} \left(\frac{T}{\tau} \right)^2} \frac{\tau^2}{T} \frac{1}{6} \left(\frac{T}{\tau} \right)^3$$

$$\leq \frac{1}{1 - \frac{T}{2\tau}} \frac{T}{6}$$

20

【0108】

DをD₀に制限するため、TTLを次のように選択すべきである。

$$D \leq D_0 \Rightarrow \frac{1}{1 - \frac{T}{2\tau}} \frac{T}{6} \leq D_0$$

$$\Rightarrow \frac{T}{6} \leq D_0 - D_0 \frac{T}{2\tau}$$

$$\Rightarrow \left(\frac{1}{6} + \frac{D_0}{2\tau} \right) T \leq D_0$$

30

$$\Rightarrow T \leq \frac{6D_0}{1 + 3\frac{D_0}{\tau}}$$

【0109】

上記の説明から、種々の値のτ及びWに対する、複数の誤り確率対TTL(T(s))の、次の表を得る。

【0110】

【表1】

40

W=15s, τ=30s			
p ₀ (%)	T(s)	p _{err} (%)	D(s)
0.1	15	0	3.3
0.5	17	0.4	4.0
1	18	0.8	4.3
5	23	4.3	6.2
10	27	7.8	8.2

【0111】

【表2】

50

$W=15s, \tau=60s$			
$\rho_0(\%)$	$T(s)$	$\rho_{err}(\%)$	$D(s)$
0.1	16	0.1	3.1
0.5	18	0.4	3.5
1	19	0.7	3.8
5	27	4.2	5.8
10	35	8.5	8.2

【 0 1 1 2 】

【 表 3 】

10

$W=15s, \tau=600s$			
$\rho_0(\%)$	$T(s)$	$\rho_{err}(\%)$	$D(s)$
0.1	19	0.1	3.2
0.5	27	0.4	4.6
1	35	0.9	6.0
5	87	4.8	15.6
10	148	9.3	28.1

【 0 1 1 3 】

【 表 4 】

20

$W=15s, \tau=3600s$			
$\rho_0(\%)$	$T(s)$	$\rho_{err}(\%)$	$D(s)$
0.1	29	0.1	4.9
0.5	62	0.5	10.4
1	99	1.0	16.7
5	389	4.8	68.5
10	749	9.3	139.3

【 0 1 1 4 】

【 表 5 】

30

$W=7s, \tau=60s$			
$\rho_0(\%)$	$T(s)$	$\rho_{err}(\%)$	$D(s)$
0.1	7	0	1.2
0.5	9	0.4	1.6
1	10	0.7	1.8
5	17	4.6	3.3
10	23	8.5	4.7

【 0 1 1 5 】

【 表 6 】

40

$W=7s, \tau=600s$			
$\rho_0(\%)$	$T(s)$	$\rho_{err}(\%)$	$D(s)$
0.1	10	0.1	1.7
0.5	17	0.5	2.9
1	23	0.9	3.9
5	73	4.8	13.0
10	133	9.3	24.9

【 0 1 1 6 】

上記の説明から、概して次の結論を導くことができる。

50

【0117】

検索した情報が「新鮮でない」(Wよりも古く)なる確率は、W以下の $T(s)$ (残存時間)の値に対しては本質的にゼロであるが、 $T(s)$ の増加に伴って、指数関数的減衰に応じて増加し、無限点に近づくに連れ100%のエラーの確率に接近する。 $T(s)$ の値がW以下に対する誤り確率がゼロであるというこの所見は、Wよりも頻繁に更新するなら情報がWより古くなることはあり得ないことは明白であるため、本質的に「些細な」ことである。しかし、Wを超える $T(s)$ の値に関しては、誤り確率の急激な増加に対して、平均更新周期 τ が、かなりの影響を与えることは明らかである。Wに対する平均更新周期 τ が大きければ大きいほど、 $T(s)$ の値の増加に対する誤り確率の増大は、それだけ緩やかである。

10

【0118】

図4は、 τ の種々の値に対するTTL(秒(s))の関数として、誤り確率 P_{err} の進展を示すグラフ400である(垂直軸がパーセンテージ(%)を示す)($W=15s$)。ライン402は $\tau=20s$ に、ライン404は $\tau=60s$ に、ライン406は $\tau=300s$ に、そしてライン408は $\tau=600s$ に対応する。

【0119】

閾値 p_0 よりも少ない誤り確率を維持するようにTTLを選択する技術は、写實的に説明するが、まず、グラフ400の垂直軸上に p_0 値を識別し、それから、グラフ400を横切って水平に想像線上を、 τ の適当な値に対応するカーブ(402、404、406あるいは408)を交差する点まで移動する。例えば、10%の誤り確率閾値を選択した場合、 $\tau=20s$ に対するTTL(ライン402)は30秒よりもやや少ない、 $\tau=60s$ に対する(ライン404)は約35秒で、 $\tau=300s$ に対する(ライン406)は約85秒で、そして $\tau=600s$ に対する(ライン408)は約150秒である。

20

【0120】

インターネット内の適当なロケーションに静的及び動的コンテンツのためのキャッシュを実装し、そして、上記の説明のように、キャッシュの動的コンテンツに対する適当な更新特性を選択することによって、動的及び静的コンテンツの両方への有効ユーザ応答時間を減少させると共に、同時に外部ネットワーク(すなわちインターネット)上の混雑を減少させることができる。

【0121】

末端ユーザ(110)が初めてウェブページを受信するとき、ウェブページは、オプションとして、ユーザのブラウザにクッキーを植えつけてもよい。その後、その末端ユーザがウェブページにアクセスするときにはいつも、ブラウザはアクセスの要求とともにクッキーを送る。クッキーは、ユーザがウェブページの検索を単に望むという事実に加えて、末端ユーザに関してサーバが知りたいと望む(すなわち必要とする)情報をサーバ(104)に知らせる。例えば、ウェブ・ページが、単一末端ユーザの個人的なウェブページである場合、サーバは末端ユーザのプリファレンスを記憶する。

30

【0122】

本発明の特徴によれば、キャッシュ(108)は、クッキーを調べて、要求のウェブ・ページが、単一末端ユーザの個人的なページであるかどうか、あるいはアクセス要求がグローバルな末端ユーザから来ていることをクッキーが示しているかどうかを調べることができる。要求ウェブ・ページが、単一末端ユーザのための個人的なウェブ・ページである(あるいは、アクセスの度にウェブ・ページが変更されるようにサポートされているということを示す他の指標がある)場合は、ウェブ・ページをキャッシュすべきではない。これは、 $TTL=0$ に設定することによって達成できる。そうでない(例えば、グローバルな末端ユーザである)場合は、TTLを単に上記の説明のように設定する。

40

【0123】

本発明は、限られた数の実施例に対して説明したが、本発明の多くの変化、変更、そして他の応用も可能である。しかし、そのような変更は、ここに開示した本発明の範囲内に含まれることは明らかである。

50

【図面の簡単な説明】

【図 1】

インターネットをかなり単純化した概要図で、本発明のキャッシングシステムの実施例を示す。

【図 2】

静的及び／あるいは動的コンテンツに対するユーザの要求が、どのように本発明によるキャッシュによって処理されるのかを示すフローチャートである。

【図 3】

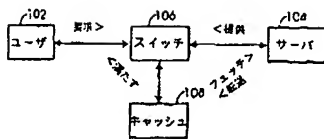
本発明の分析による、平均誤り確率を示すグラフである。

【図 4】

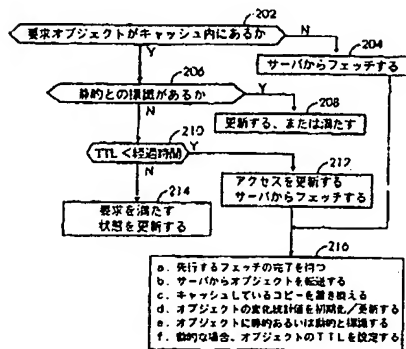
本発明の分析による、誤り確率の進展を示すグラフである。

10

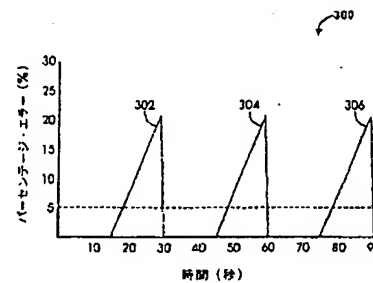
【図 1】



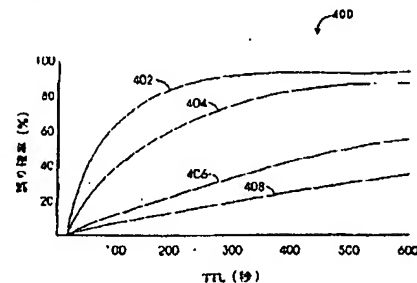
【図 2】



【図 3】



【図 4】



【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(14) World Intellectual Property Organization
International Bureau(43) International Publication Date
24 January 2002 (24.01.2002)

PCT

(10) International Publication Number
WO 02/07364 A2

- (51) International Patent Classification: H04L
- (21) International Application Number: PCT/IL01/00651
- (22) International Filing Date: 16 July 2001 (16.07.2001)
- (23) Filing Language: English
- (24) Publication Language: English
- (26) Priority Data: 60218559 (17 July 2000 (17.07.2000)) US
- (71) Applicant (for all designated States except US): KPLICA-
TION NETWORKS LTD. (IL/LU); 38 Hamaar St., 45241
Tel Hasharon (IL).
- (72) Inventors: and
(73) Inventor/Applicant (for US only): MELAMED,
Shmuel (IL/LU); 53 Zarnit St., 45350 Tel Hasharon (IL).
BIGIO, Yoram (IL/LU); 12 Barzani St., 69121 Tel Aviv
(IL).
- (74) Agent: FRIEDMAN, Mark, M., Ral Samuchoff, 7 Ham-
mation St., 67897 Tel Aviv (IL).
- (81) Designated States (national): AR, AU, AT, AM, AI, AF,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,
CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GR, GM,
HN, HU, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LA, LK,
LU, LY, LT, LV, MA, MD, ME, MG, MN, MW, MX,
MZ, NA, NZ, NL, PT, RO, RU, SD, SE, SG, SI, SK, SL,
TH, TJ, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LI, NL, NO, PL, PT, SE, TR), OAPI patent (BF, BJ, CF,
CG, CI, CM, GA, GN, GW, ML, MR, NE, NG, SN, TD, TG).
- Published:
without international search report and to be republished
upon receipt of that report
- For two-letter codes and other abbreviations, refer to the "Guide-
ance Notes on Filing and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.



WO 02/07364 A2

(54) Title: BANDWIDTH SAVINGS AND QoS IMPROVEMENT FOR WWW SITES BY CACHING STATIC AND DYNAMIC
CONTENT ON A DISTRIBUTED NETWORK OF CACHES

(57) Abstract: Caches are deployed in the Internet for storing and updating copies of objects having dynamic content. Update characteristics of the objects are determined, and a time to live (TTL) parameter for the objects is adjusted based upon the update characteristics. Generally, the object is updated if its TTL is less than its age. The TTL for an object may be adjusted to (i) maintain its probability of error below a predetermined error probability threshold, (ii) maintain its error rate below a predetermined error probability threshold, or (iii) maintain its delay time below a predetermined delay threshold. Preferably, the caches are dedicated machines and are placed so that Web browser queries through the cache instead of going all the way to the original sites, in many different locations, ideally within the network of ISPs providing the Internet connectivity to the highest number of users in those locations. In this manner, the users of those ISPs and, in a lesser extent, neighboring ISPs, will enjoy a large QoS and speed improvement, for most of the traffic will stay within or close to the ISPs' internal networks and need not go through the highly-loaded Internet backbone, and the original web sites will no longer need as much bandwidth, since the caches will absorb most of the load. The system can adapt, in real time, according to the number of requests to each page and the actual update frequency of the page.

WO 02/07364

PCT/IL01/00651

**BANDWIDTH SAVINGS AND QoS IMPROVEMENT FOR WWW SITES
BY CACHING STATIC AND DYNAMIC CONTENT ON A DISTRIBUTED NETWORK
OF CACHES**

5 TECHNICAL FIELD OF THE INVENTION

The invention relates to the storage and transport of information on an Internet and, more particularly, to caching Web-page content on the Internet.

CROSS-REFERENCE TO RELATED APPLICATION(S)

10 This is a continuation-in-part of commonly-owned, copending U.S. Provisional Patent Application No. 60/218,559, filed 17 July 2000.

BACKGROUND OF THE INVENTION

15 When two or more computers are connected so that they can share resources, this is referred to as a "network". When two or more networks are connected together, this is referred to as an "internet" (lower case 'i'). The "Internet" (upper case 'I') is a global internet, which is a plurality of inter-connected networks, all using a common protocol known as TCP/IP (Transmission Control Protocol/Internet Protocol).

20 The World Wide Web ("WWW", or simply "Web") is a set of standards and protocols that enable the exchange of information between computers, more particularly hypertext (HTTP) servers, on the Internet, tying them together into a vast collection of interactive multimedia resources. Traffic across the Internet passes through multiple "backbone" carriers, such as UUNET or Digex, which forward the signals to one another under a system called peering.

25 "Quality of Service" (QoS) is a measure of the speed and reliability with which information can be transported from a content source (typically a server, or an Internet Service Provider, ISP) to a user's computer (upon which is typically running a type of "browser" software that is used to view content). "Bandwidth" is a measurement of the volume of information that can be transmitted over a network (or a given portion thereof) at a given time. The higher the bandwidth, the more data can be transported. Latency is a measure of the time that it takes for a packet of data to traverse the Internet, from source to user.

30 Bandwidth can be increased by installing new capital resources, but increasing bandwidth only indirectly addresses latency, and increasing bandwidth cannot accelerate overloaded or slow origin servers. Other solutions are needed to overcome these barriers. One solution, caching content, is discussed at length in an article entitled "Network Caching Guide, Optimizing Web Content Delivery", by Michael A. Gould, March 1999, which is available online (via the Internet) at the following address:

40 http://www.jtktoni.com/products/network/tech_resources/tech/cachingguide.pdf
A text version of the article is available online at the following address:
http://www.google.com/search?q=cache:www.jtktoni.com/products/network/tech_resources/tech/cachingguide.pdf

45 As noted in the Gould article:

50 "One way to improve performance and make it more predictable is to minimize the variability introduced by long trips across the Internet from the content source to the user's browser. By storing frequently accessed content at a location closer to the user, a great deal of latency and unpredictable delay is

WU 0207364

PCT/JP01/00651

2

the Internet can be eliminated. The technique for doing this is called caching. Caching is a means of storing content objects from a Web server closer to the user, where they can be retrieved more quickly. The storehouse of objects, including text pages, images, and other content, is called a Web cache (pronounced cash)." (Gould, page 2)

Caching is an approach that improves QoS for users while improving the economics for network service providers. Caching entails storing frequently accessed Web content closer to users, thereby reducing the number of hops that must be traversed in order to retrieve content that resides on a remote site. (Gould, page 8)

In operation, requests for content originate from a user's browser and are first directed to the caching server. If the content is currently contained in the cache and is up to date, the content is sent to the user without the browser's request having to be sent on to the originating server. This can reduce the latency for the transfer of content and reduce the amount of time it takes to transfer the content to the user. It also reduces the amount of data the service provider has to retrieve upstream from the Internet to fulfill requests. (Gould, page 8)

Caching is a concept that is well understood in computer hardware design. Modern microprocessors employ on-chip caches of high-speed memory to store frequently used instructions and data locally instead of having to read them from slower memory. Computers implement several additional levels of caching, including RAM cache and even on-disk cache, all designed to reduce the latency for reading instructions and data and speeding up the transfer of data within the system. The basic principle behind caching is to store frequently accessed data in a location that can be accessed more quickly than it could from the data's more permanent location. In the case of a system's CPU, the permanent location is the disk. On the Web, the permanent location is the origin server somewhere out on the Internet. (Gould, page 8-9)

Network caching, while a newer concept, is based on exactly the same principle and has the same benefit of cost-effectively improving the user experience. (Gould, page 3)

"Caching provides direct benefits to the end user in terms of reduced latency of Web page download (the time you have to wait before anything starts to happen) and faster download speeds (the time it takes for all the downloading to finish). But caching also provides benefits to network service providers. By storing, or caching, Web content that users request from remote servers locally, fewer of these requests have to be sent out over the Internet to be fulfilled. As a result, the access provider maintaining a cache has its upstream bandwidth requirements reduced. This reduces the bandwidth the service provider has to purchase in order to provide its customers with a satisfactory Web experience. (Gould, page 3)

"Caching also provides several benefits in a Web hosting environment in which an access provider hosts Web sites for many customers. It can reduce the impact of traffic spikes caused by high-interest content and also serve as

WU 0207364

PCT/1101/00651

3

the basis for a variety of value-added services, such as providing additional capacity, guaranteed service levels, and replication services. (Gould, page 3)

5 "[Caching] network data makes it possible to provide users with an optimal experience and predictable response times. With typical cache hit rates, the user experience has a higher quality of service (QoS). Improved QoS provides significant benefits to ISPs, content providers, and corporate sites. Better QoS results in higher customer loyalty and retention. It helps create a stronger brand equity, both for the access provider and for content providers. Content providers who ensure that their content is simply cached throughout the network, close to users, will ultimately see more visitors accessing their content. (Gould, page 5)

10 "Web browsers also implement a form of caching, storing recently accessed Web content on the user's hard disk and reading from that local cache of files instead of accessing the Internet. This works well when the user hits the "Back" and "Forward" buttons in their browser during a session, but it does nothing if the user is browsing a site for the first time. (Gould, page 9)

15 "Neither the browser's cache nor a Web server's cache can address network performance issues. By placing a cache of Web content on the network between the user and the originating Web sites, the distance that commonly accessed content has to travel over the Internet is reduced, and users experience quicker response and faster performance. Network caching takes advantage of the fact that some content is accessed more frequently than other content. By implementing a caching solution, network service providers can provide a better Web experience to their customers while reducing the total requirement for high-bandwidth connections to the Internet. (Gould, page 9)

20 "When a caching solution is not in place, requests for content from a browser and the content delivered from the origin server must repeatedly traverse a complete trip from the requesting computer to the computer that has the content. The Web browser sends a request for a Uniform Resource Locator (URL) that refers to a specific Web page on a particular server on the Internet. The request is routed to the server through the normal TCP/IP network transport. The content requested from the Web server (also known as an HTTP server) may be a static HTML page with links to one or more additional files, including graphics. The content may also be a dynamically created page that is generated from a search engine, a database query, or a Web application. The HTTP server returns the requested content to the Web browser one file at a time. Even a dynamically created page often has static components that are combined with the dynamic content to create the final page. (Gould, page 13)

25 "When caching is used, frequently accessed content is stored close to the user. It may be stored inside a firewall on the user's corporate LAN, at the user's ISP, or at some other Network Access Point (NAP) or Point of Presence (POP) located closer to the user than the majority of Web servers. If the requested content, or file, is located in the cache and is current, or fresh, that is considered a cache hit. The more frequently user requests can be served from

WU 0207364

PCT/JP01/00651

these caches, the higher the hit rate, the better the user's performance, and the less data the service provider has to retrieve from the Internet on behalf of the user. (Goulds, page 14)

Similar issues exist for FTP file transfers, with an FTP server handling each request for a file submitted by the FTP client application. Delays and bottlenecks can be a bigger problem with FTP because the typical size of a file transferred by FTP is larger than a typical HTML file. Streaming audio and video are additional examples of an Internet application that can benefit from caching content close to end users. (Goulds, page 14)

Network caching can be applied to content delivered over many different protocols. These include HTTP, NNTP, FTP, RISP, and others. All are characterized by having some proportion of static content and high utilization. Cache server support for each protocol is, of course, required." (page 14)

"Since the cache must be kept fresh, there will still be traffic from the ISP out to the Internet, even if every bit of content requested by users is in a cache. Page freshness has to be assured, and new content must be downloaded. But by using caching, bandwidth utilization can be optimized. It is even possible to use a cache with dynamic content, since even these pages have some static content that can be served from a cache. Depending on the distribution of traffic and the scalability of the cache, up to 40 percent of user HTTP requests can be taken off the network and served from a cache. This makes networks more efficient, enabling better performance to be offered at lower cost." (Goulds, page 14)

The Gould article discusses the effective use of caches, load balancing, where to locate caches in the infrastructure, and designing cache-friendly web content. There is also mention of protocols which have been developed - for example, Web Cache Control Protocol (WCCP) (page 18). There is also discussion of appropriate use of the "expire" and "max-age" headers in the HTTP protocol. (Goulds, page 27). And, as expressly stated by Gould,

"With good design even dynamically generated pages can benefit from caching. By separating the dynamic content of a page from the static content, the static content can be cached and the dynamic content retrieved and downloaded separately. (Goulds, page 28; emphasis supplied)

It is therefore known to cache static data from Web sites using a dedicated computer's RAM and hard disk, so that this computer can act as a proxy between the WWW ("web") servers and their users. Several requests for a given Web page can then be served at the cost of a single request to the original Web server. Internet Service Providers (ISPs) commonly provide such a caching service to their customers. However, this technique suffers from two main drawbacks:

- it is not applied globally, and even users of the Web sites who have access to a cache have to deliberately activate this feature, of which they are often not aware; and
- it does not apply to pages which change very often, as is the case with dynamic content (content which is generated on-the-fly by the remote server).

There therefore exists a need for the caching of dynamic content, as well as static content.

WU 0207364

PCT/JP01/00651

5

Glossary/Definitions

Unless otherwise noted, or as may be evident from the context of their usage, any terms, abbreviations, acronyms or scientific symbols and notations used herein are to be given their ordinary meaning in the technical discipline to which the invention most nearly pertains. The following glossary of terms is intended to lend clarity and consistency to the various descriptions contained herein, as well as in any prior art documents which may be cited:

- 10 Cache Server A highly optimized application that stores frequently accessed content at strategic aggregation points close to the users requesting that content in order to reduce the impact of delays and network bottlenecks.
- 15 C-ARP Cache Array Routing Protocol. A protocol for synchronizing multiple cache servers. C-ARP maintains a shared namespace that maps any Web object's address (URI) to only one node in the array. Requests are routed to that node.
- 20 Cookie The most common meaning of "Cookie" on the Internet refers to a piece of information sent by a Web Server to a Web Browser that the Browser software is expected to save and to send back to the Server whenever the browser makes additional requests from the Server. Depending on the type of Cookie used, and the Browser's settings, the Browser may accept or not accept the Cookie, and may save the Cookie for either a short time or a long time. Cookies might contain information such as login or registration information, online "shopping cart" information, user preferences, etc. When a Server receives a request from a Browser that includes a Cookie, the Server is able to use the information stored in the Cookie. For example, the Server might customize what is sent back to the user, or keep a log of particular user's requests. Cookies are usually set to expire after a predetermined amount of time and are usually saved in memory until the Browser software is closed down, at which time they may be saved to disk if their "expire time" has not been reached.
- 25 30 35 Dynamic Content "Live" content which is updated on a regular basis. Examples of dynamic content might include a "current temperature" display on a weather web site, search results, or a "Current Top Headlines" item on a news web site.
- 40 HTTP Server A server that implements the HTTP protocol, enabling it to serve Web pages to client agents (browsers). HTTP Servers support interfaces so that Web pages can call external programs. They also support encryption mechanisms for securely exchanging information and authentication and access control mechanisms to control access to content.
- 45 50 ICP Internet Cache Protocol. A protocol for synchronizing multiple cache servers. Each time a cache server experiences a miss, it broadcasts messages to all peer nodes asking whether each has the content. The

WU 0207364

PC/T/11/01/00651

6

requesting server then must issue a request for the content and forward it on to the user.

5	Proxy Server	A proxy server acts as an intermediary between a user and the Internet so an enterprise can ensure security and administrative control and also provide a caching service. A proxy server is usually associated with or part of a gateway server that separates the enterprise network from the outside network or a firewall that protects the enterprise network from outside intrusion.
10	Routers	Routers are the devices that build a fully interconnected network out of a collection of point-to-point links. Routers on the Internet exchange information pertaining to their local section of the network, particularly how close they are topologically to local systems. They collectively build a map of how to get from any point in the Internet to any other. Packets are routed based on the exchanged mapping information, until the last router connects directly to the target system.
15		
20	Static Content	"Fixed" or long-term unchanging components of web pages stored as files that are either never changed or are changed only on an infrequent basis.
25	Switches	High-speed network devices that typically sit on the periphery of the Internet. Switches differ from routers in providing higher performance at a lower price but with limited functionality. Typical switches can route traffic locally but aren't concerned with complexities of routing found in the high-speed Internet backbone. Switches play an important role in caching because they are often used to divert the cacheable traffic to the caching system.
30	HTML	Hypertext Markup Language. A specification based on Standard Generalized Markup Language (SGML) for tagging text so that it may be displayed in a user agent (browser) in a standard way.
35	HTTP	Hypertext Transmission Protocol. An application-level protocol that runs on top of TCP/IP, which is the foundation for the World Wide Web.
40	IP	Internet Protocol. The network layer for the TCP/IP protocol suite. It is a connectionless, best-effort, packet-switching protocol.
45	IP Address	A 32-bit address defined by the Internet Protocol that is usually represented in decimal notation. It uniquely identifies each computer on the Internet.
50	Protocol	An agreed-upon set of technical rules by which computers exchange information.
	URI.	Uniform Resource Locator. The method by which Internet sites are addressed. It includes an access protocol and either an IP address or

WU 0207364

PCT/11.01/00651

7

DNS name. An example is <http://www.domain.com>.

Usenet

5

Short for User's Network. A collection of tens of thousands of bulletin boards residing on the Internet. Each contains discussion groups called newsgroups dedicated to various topics. Messages are posted and responded to over the Network News Transfer Protocol (NNTP).

Web Server

See HTTP Server.

10

WU 0207364

PCT/IL01/00651

BRIEF DESCRIPTION OF THE INVENTION

An object of the invention to provide a technique for reducing bandwidth usage of WWW servers and improving the QoS of WWW sites.

5 According to the invention, a technique for caching objects having dynamic content on the Internet generally comprises disposing a cache in the Internet for storing and updating copies of dynamic content. The cache may be disposed at a location selected from the group consisting of major Internet switching locations, dial-in aggregation points, and corporate gateways. The cache may also store and update copies of static content.

According to a feature of the invention, update characteristics of the objects are determined, and a time to live (TTL) parameter for the objects is adjusted based upon the update characteristics. Generally, the object is updated if its TTL is less than its age. The TTL for an object may be adjusted to:

- (i) maintain its probability of error below a predetermined error probability threshold;
- (ii) maintain its error rate below a predetermined error probability threshold; or
- (iii) maintain its delay time below a predetermined delay threshold.

10 According to the invention, a method of responding to a user request for an object having dynamic content, said object originating from a server, comprises storing a copy of the object in a cache; establishing a time to live (TTL) for the object; receiving the user request at the cache;

15 fulfilling the user request with the stored copy of the object if its TTL is greater than its age; and fetching an updated copy of the object and responding to the user request with the updated copy if the TTL of the stored copy is less than its age.

According to a feature of the invention, the TTL for the object is first set to a reasonable lower limit (T_{min}) and is then adjusted based on the frequency at which the object actually changes.

According to a feature of the invention, each time the cache fetches the object from the server, the cache performs the following procedures:

- a. if another fetch for the same object is ongoing, waiting for the previous fetch to complete;
 - b. fetching the object from the server;
 - c. replacing the cached copy, if present, by the fetched object, after having compared them to determine whether the object had changed since it was last fetched;
 - d. initializing or updating the object's change statistics accordingly;
 - e. marking the object as static or dynamic content depending on the server's reply; and
 - f. if the object is dynamic, setting its TTL (T) to an appropriate value with respect to an average time between changes (τ) determined from the object's change statistics, the number of user requests per time unit (η) determined from the object's access statistics, and one of the following procedures (A-E):
- A. maximum error probability;
 - B. maximum error rate;
 - C. maximum delay;
 - D. any combination of the above (A,B,C), taking the lowest T ; or
 - E. any combination of the above (A,B,C), but keeping T within a predetermined window of "reasonable" values bounded by T_{min} and T_{max} .

WU 02/07364

PC/T/1.01/00651

9

Preferably, the caches are dedicated machines and are placed so that Web browsing passes through the cache instead of going all the way to the original sites, in many different locations, ideally within the network of ISPs providing the Internet connectivity to the highest number of users in those locations. In this manner:

- the users of those ISPs and, to a lesser extent, neighboring ISPs, will enjoy a huge QoS and speed improvement, for most of the traffic will stay within or close to the ISPs' internal networks and not need to go through the highly-loaded Internet backbone; and
- the original web-sites will no longer need as much bandwidth, since the caches will absorb most of the load.

Since many web-sites tend to serve all-dynamic content, dynamic content will be cached, as well as static content. However, it is not merely a matter of the caches remembering the content, as there is no indication as to when the latter is going to change. Hence, it needs to be reloaded, periodically, instead of simply serving the cached copy.

Therefore, the cache reloads a page (fetches an object from the server), whenever its corresponding cached copy has not been refreshed for a given time (time to live, "TTL"). TTL can be thought of as the "shelf life" of the page. The cache first sets the TTL for a dynamic object to a reasonable lower limit T_{min} . Then, over time, as the cache reloads the page several times, it can track the frequency at which the page actually changes content, and adjust the TTL for that page accordingly. This "TTL" technique mimics a common caching method for static pages, if the original server of the page specifies a TTL. But since servers for dynamic pages do not specify a TTL for them, the cache has to establish a reasonable TTL of its own accord.

Therefore:

- whole web-sites can be cached, including their dynamic content;
- traffic between caches and the original sites is better managed;
- while some users might get an outdated content, in the case of a page changing before the cache reloads it, it is possible to limit the delay between the two (which could happen anyway without the cache, when the network gets saturated) as well as the error probability;
- the system can adapt, in real time, according to the number of requests to each page and the actual update frequency of the page.

Other objects, features and advantages of the invention will become apparent in light of the following description thereof.

40

WO 02/07364

PCT/JP01/00651

10

BRIEF DESCRIPTION OF THE DRAWINGS

Reference will be made in detail to preferred embodiments of the invention, examples of which may be illustrated in the accompanying drawing figures. The figures are intended to be illustrative, not limiting. Although the invention is generally described in the context of these preferred embodiments, it should be understood that it is not intended to limit the spirit and scope of the invention to these particular embodiments.

In flowcharts presented herein, rectangular boxes generally represent a sequential step being performed, a generally rectangular box with pointed ends represents a decision step (test) having two mutually-exclusive results ("Y"=Yes, "N"=No), and an empty circle is not a step or a test, but is merely a graphical junction point at which two or more paths in the flowchart converge.

The structure, operation, and advantages of the present preferred embodiment of the invention will become further apparent upon consideration of the following description, taken in conjunction with the accompanying figures, wherein:

Figure 1 is a greatly simplified schematic illustration of the Internet, illustrating an embodiment of the caching system of the present invention;

Figure 2 is a flowchart illustrating how user requests for static and/or dynamic content are handled by the cache, according to the invention;

Figure 3 is a graph illustrating an average error probability, according to an analysis of the invention; and

Figure 4 is a graph illustrating the evolution of error probability, according to an analysis of the invention.

WU 0207364

PCT/11/01/00651

11

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is a greatly simplified schematic illustration of the Internet environment, illustrating an embodiment of the caching system of the present invention. Generally, a user 102 makes a "request" for an "object" (e.g., a Web page) which is made available on the Internet by a server (e.g., ISP) 104. (The object typically originates at a content provider, not shown.) A switch 106 interfaces a cache (cache server) 108 to the Internet. The cache may contain a copy of the Web page. There are two possible "responses" to the user request - either the server "serves" (or "serves") the request, or it is "fulfilled" in the cache. In the latter case, the content must first have been "transferred" to the cache, which may periodically "fetch" (or "reload") updated Web page content from the server. These terms will be adhered to, to the extent possible, in the discussion that follows.

As discussed in the Gould article (e.g., page 18, Illustration 4), the switch 106 may be a high-performance processor that can look at network traffic and make routing decisions based on protocols above the IP level. As a result, the switch can direct HTTP (and other) traffic to caches (108), and send the rest of the traffic directly to the Internet. This is exemplary of one of a number of possible system architectures. Which architecture is used depends on several factors, including where the cache is implemented, the primary purpose of the cache, and the nature of the traffic.

As further discussed in the Gould article:

"Caches can either be deployed in a transparent or nontransparent form. A nontransparent cache is explicitly visible, and browsers or other caches that use the cache are overtly configured to direct traffic to the cache. In this case, the cache acts as a proxy agent for the browser, fulfilling requests when possible and forwarding requests to the origin server when necessary. Nontransparent caches are often a component of a larger proxy server acting as part of a gateway or firewall and addressing many different applications. (Gould, pages 16-17)

A transparent cache sits in the network flow and functions invisibly to a browser. For ISP and enterprise backbone operations, a transparent configuration is preferred because it minimizes the total administrative and support burden of supporting users in configuring their browsers to find the cache. (Gould, page 17)

Caches should be implemented transparently to maximize the benefits of caching. A nontransparent implementation requires having browsers manually configured to direct their requests for content to the cache server. In a transparent configuration, cache benefits are delivered to clients without having to reconfigure the browser. Users automatically gain the benefits of caching. (Gould, page 17)

In an enterprise environment, transparency can be implemented either through automatic browser configuration or by intercepting traffic on the network. Both Netscape and Microsoft provide utilities for centrally configuring large networks of browsers and for customizing browser configurations being installed on users' PCs. Browser plug-ins can also provide automatic

configuration. Although this approach is transparent to the user, it does require administrative effort on an ongoing basis. (Gould, page 17)

There are several options for providing transparent caching:

- The cache can be configured as if it were a router so that all Internet-based traffic is aimed at it. This is a transparent configuration that requires no configuration of the browser; the browser or downstream cache is unaware of the cache's existence but still benefits from it. The downside is that the system on which the cache resides has to devote some of its resources to routing, and the cache becomes a mission-critical part of the network. Sophisticated router configuration with policy-based routing can minimize some of these issues by only directing HTTP (TCP Port 80) traffic to the cache, bypassing the cache in the event of failure and sending traffic directly to the Internet. (Gould, page 17)

- An increasingly popular option is to use a Layer 4 switch to interface the cache to the Internet (see Illustration 4). These switches, currently offered by Alteon, Foundry, ArrowPoint, and others, are high-performance processors that can look at network traffic and make routing decisions based on protocols above the IP level. As a result, the switch can direct HTTP (and other) traffic to the caches and send the rest of the traffic directly to the Internet ... [I]f the switch can parse the HTTP request and send the request to a specific node in a cache farm based on the URL requested. Using an intelligent switch keeps unnecessary network traffic off the cache, simplifies designing for availability, and distributes loading on the cache farm based on specific URLs. (Gould, page 18) (An architecture similar to this one is described hereinabove with respect to Figure 1)

- Another option for transparency is the Web Cache Control Protocol (WCCP). WCCP was developed by Cisco Systems to allow Web caches to be transparently installed in networks using Cisco IOS-based routers. With WCCP, HTTP traffic is redirected to the Web cache instead of the origin server. WCCP does not require any changes to the network architecture, thereby simplifying the implementation of transparent Web caching. (Gould, page 18)

- Web Proxy Autodiscovery Protocol (WPAD) is a new proposed standard protocol, which, when integrated with browsers, streaming media clients, and other Internet client software, is designed to automatically locate caches and services on the network without requiring any configuration by end users. WPAD provides a flexible, vendor-neutral software alternative to existing cache transparency solutions that utilize routing or switching equipment. In the future, WPAD-enabled client software will automatically connect users with embedded network services in their region, providing simplicity for both users and the network providers that deploy these services. (Gould, pages 18-19)

Caching systems can be used to optimize the performance of a Web server site as well as to speed Internet access for Web browser users. In a reverse proxy configuration, the caching system sits in front of one or more Web servers, intercepting traffic to those servers and standing in, or proxying, for one or more of the servers. Cache servers can be deployed throughout a network, creating a distributed network for hosted content. When necessary the proxy

WU 0207364

PC1771.DL00651

13

cache server will request dynamic and other short-lived content from the origin servers. This enables content from the site to be served from a local cache instead of from the origin server. The proxy server can be optimized for high performance, efficient operation, conserving resources, and off-loading the origin server from serving static content. Reverse proxy caching provides benefits to the access provider as well as to the user. Those benefits include the ability to enable load balancing, provide peak-demand insurance to assure availability, and provide dynamic mirroring of content for high availability. (Gould, page 20)

There are three general characteristics that describe where caches are best located on a network:

- Choke Points in the Network Traffic. There are locations where a large majority of network traffic passes and is therefore visible to the cache server. This allows the cache to handle more requests and store more content than if located somewhere that can be easily bypassed.
- Points with High-Network Load. High traffic conditions allow higher cache utilization and therefore greater benefits can be achieved.
- Locations that Produce Greatest Economic Benefits for a Cache. Points where users will benefit from high cache hit rates while also reducing upstream bandwidth requirements will provide both QoS benefits and economics for the access provider. Locations with these characteristics are typically found at major Internet switching locations, dial-in aggregation points, or corporate gateways. (Gould, page 20)

Locations with these characteristics are typically found at major Internet switching locations, dial-in aggregation points, or corporate gateways, including:

- POP and DIAI-UP (see Gould, page 21, Illustration 7)
- NAPS (see Gould, page 22, Illustration 8)
- Satellite (see Gould, page 22)
- International Gateway (see Gould, page 23, Illustration 9)
- Web hosting (see Gould, page 24, Illustration 10)
- Last Mile (see Gould, page 25, Illustration 11)
- Corporate Gateways (see Gould, page 25, Illustration 12)

A person having ordinary skill in the art will readily understand where to locate the cache(s) of the present invention in light of the description set forth hereinabove. Generally speaking, the cache of the present invention can be located anywhere that there is (or could be) a cache serving static content, or it can be incorporated into an existing cache which fulfills requests for static content, with the additional functionality enabled according to the techniques set forth below. Or, it can be provided as a separate, dedicated machine (computer).

Figure 2 is a flowchart illustrating how user requests for static and/or dynamic content are handled by the cache.

In a first step 202, for a user request for an object, the cache determines whether the requested object is in cache. If not (N), the user request is passed on to the server for servicing the request and meanwhile, in a step 204, the cache fetches the object from the server in anticipation of the next request for the object from the same or another user.

WU 0207264

PCT/JP01/00651

14

Generally, in this example, all requests for objects are presumed to go through a cache server, which is transparent to the user. It intercepts information requests and decides whether it will provide a response from a cached local copy or from a remote information source. After fetching information from a local source, the cache server decides whether to store it locally, and if so, for how long. A request for information which can be provided from a local copy is known as a "cache hit". Conversely, a request for information which is not stored locally is known as a "cache miss". When the storage determination algorithm is well-designed the probability of a cache hit is greatly improved, and apparent response time to user requests (related to QoS) is reduced. Further, every information request satisfied by locally cached content (cache hit) reduces traffic on the external network, permitting shorter response times over the external network.

If the requested object is in the cache, it is next determined in a step 206 whether the requested object is marked as static. If so (Y), it is then determined in a step 208 whether to update the cached copy or to use it to fulfill the user request, using any suitable standard algorithm for caching static objects, such as comparing the objects "age" (the time elapsed since it has last been refreshed) to the TTL (if the original server of the page specifies a TTL), asking the server the latest modification time, etc).

If the requested object is in cache, and it is dynamic (N, step 206), it is determined in a step 210 whether the cached copy's TTL ("shelf life") is less than its age.

If the cached copy's TTL is less than (a lower number than) its age (Y, step 210), it is considered to be "stale", and in a step 212 the cache:

- updates the object's access statistics (number of user requests per time period - last few minutes, last hour, etc...); and
- fetches the object from the original server.

If the cached copy's TTL is equal to or greater than its age (N, step 210), it is considered to be "fresh", and in a step 214 the cache:

- fulfills the request using the cached copy; and
- updates the object's access statistics.

Optionally, if the time difference between the cached copy's age and its TTL is less than a given time, and the number of recent user requests is more than a given rate, it is considered to be "aged" and "popular", and the cache fetches the object from the server in what is termed an "anticipated refresh".

Additionally, for each time the cache fetches an object from the server (see steps 204 and 212), the following procedures are performed by the cache, in a step 216:

- a. if another fetch for the same object is ongoing (e.g., due to a previous user request), the cache waits for the previous fetch to complete, rather than duplicating it request;
- b. fetches the object from the server;
- c. replaces its cached copy, if present, by the up-to-date object, after having compared them to determine whether the object had changed since it was last fetched;
- d. initialize or update the object's change statistics (number of changes per time period) accordingly;
- e. mark the object as static or dynamic content depending on the original server's reply (presence of a modification time, or of a nonzero TTL, etc.);

WO 0207364

PCT/11/01/00651

15

f. if the object is dynamic, set its TTL (T) to the appropriate value with respect to the average time between changes τ (determined from the object's change statistics), the number of user requests per time unit n (determined from the objects access statistics), and a selected one of the following procedures (A-E):

A. maximum error probability (p_0), which is the average ratio of the number of requests fulfilled using a cached copy whose corresponding original object has changed for more than a given time W , over the total number of requests:

$$T = (W + \tau p_0) \left(1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right)$$

B. maximum error rate (n_0), which is the average number per time unit of requests fulfilled using a cached copy whose corresponding original object has changed for more than a given time W :

$$T = \left(W + \tau \frac{n_0}{n} \right) \left(1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right)$$

C. maximum delay (D_0), which is the average time between an object change and when the cached copy is refreshed:

$$T = \frac{6 D_0}{1 + 3 \frac{D_0}{\tau}}$$

D. any combination of the above (A,B,C), taking the lowest T ; or

E. any combination of the above (A,B,C), but keeping T within a predetermined window of "reasonable" values bounded by T_{min} and T_{max} .

In the above equations, the following parameters have been used:

- T is Time To Live for the dynamic object;
- W is a given time since the original object has changed (i.e., how long it is outdated);
- τ is an average time between changes, which is determined from the object's change statistics;
- n is number of user requests per time unit (e.g., frequency);
- p_0 is maximum error probability, which is the average ratio of the number of requests fulfilled using a cached copy whose corresponding original object has changed for more than the given time W , over the total number of requests;

WU 0207364

PCT/M.01/00651

16

- η_k is maximum error rate, which is the average number per time unit of requests fulfilled using a cached copy whose corresponding original object has changed for more than the given time M_k .
- D_0 is maximum delay, which is the average time between an object change and when the cached copy is refreshed.

Selecting and Adjusting TTI.

A simple model, which takes no other parameter than the average time between changes (τ) of the object is the exponential decrease model - that is, the probability that the object does not change during a time period of length t is considered to be of the form $e^{-t/\tau}$. It is, of course, 1 for $t=0$, and tends to 0 as t tends to $+\infty$.

Considering a situation with many user requests, as compared to the refresh rate of the cached copy, so the cache will update at regular intervals, despite being user-driven.

Average frequency of changes

The model's consistency is now checked, especially whether the average update frequency of a given object is $1/\tau$. Let $p_k^*(t)$ be the probability that k changes over a time period of t . By construction:

$$p_0^*(t) = e^{-t/\tau}$$

And, for $k > 0$, $p_k^* = \int_0^t p_{k-1}^*(u)(1 - e^{-t-u/\tau})e^{-t-u/\tau} du = \int_0^t p_{k-1}^*(u) \frac{du}{\tau} e^{-(t-u)/\tau}$

(the probability that k changes occur over $[0:t]$ is the sum over u of the probability that $k-1$ changes occurred between 0 and u , that there was a change between u and $u+du$, and none between $u+du$ and t).

One can deduce by recurrence that:

$$p_k^*(t) = \frac{1}{k!} \left(\frac{t}{\tau} \right)^k e^{-t/\tau}$$

$$\text{as: } \int_0^t \frac{1}{k!} \left(\frac{u}{\tau} \right)^k e^{-u/\tau} \frac{du}{\tau} e^{-(t-u)/\tau} = \frac{1}{k! \tau^{k+1}} e^{-t/\tau} \int_0^t u^k du$$

$$= \frac{1}{k! \tau^{k+1}} e^{-t/\tau} \frac{t^{k+1}}{k+1}$$

$$= \frac{1}{(k+1)!} \left(\frac{t}{\tau} \right)^{k+1} e^{-t/\tau}$$

WU 02/07364

PCT/11/01/00651

17

Now the average number of changes over a time period of t is (by definition of averaging):

$$\begin{aligned}
 \sum_{k=0}^{\infty} k \times p_k(t) &= \sum_{k=1}^{\infty} k \times p_k(t) \\
 &= \sum_{k=1}^{\infty} k \times \frac{1}{k!} \left(\frac{t}{\tau}\right)^k e^{-t/\tau} \\
 &= e^{-t/\tau} \times \sum_{k=1}^{\infty} \frac{1}{(k-1)!} \left(\frac{t}{\tau}\right)^{k-1} \\
 &= e^{-t/\tau} \times \left(\frac{t}{\tau}\right) \times \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{t}{\tau}\right)^k \\
 &= e^{-t/\tau} \left(\frac{t}{\tau}\right) e^{t/\tau} \\
 &= \frac{t}{\tau}
 \end{aligned}$$

- 5 Hence, the changes do have an average frequency of $1/\tau$.

Average error probability

- The error probability is defined as being the percentage of cases where the user receives the cached copy whereas the object has changed on the original server, and the cached copy is outdated by more than a given time W . Figure 3 illustrates the percentage error (vertical axis, in percentages) versus time (horizontal axis, in seconds) for instance, with $W=15$ s, $\tau=60$ s, and $T=30$ s. The graph illustrates the probability that the cached copy is "stale" by more than W seconds. In the graph, W is 15 seconds, so the probability is zero between 1 and 15 seconds. Then, the probability rises until 30 seconds, when a re-fetch occurs ($T=30$ s), as illustrated by the sawtooth pattern 302. This pattern repeats itself every 30 seconds. Between 30 and 45 seconds, the probability is again zero, and a re-fetch occurs at 45 seconds, and the probability rises until 60 seconds, as illustrated by the sawtooth pattern 304. A similar result is shown by the sawtooth pattern 306 between 75 and 90 seconds.

- 20 Given that the cache is supposed to update the copy of the object at regular intervals, the calculations can be performed over a time interval $[0; T]$.

- The average time E within the interval during which the content is outdated is the sum over all intervals $[u; u+du]$ of the probability that there was no change between 0 and u , but that there was a change between u and $u+du$, multiplied by the length of time during which the content remains outdated by more than W . That is:

WU 0207364

PCT/JP01/00651

13

$$E = \int_0^{T-W} e^{-u/\tau} \frac{du}{\tau} (T-W-u)$$

$$= (T-W) \left(1 - e^{-\frac{T-W}{\tau}}\right) - \tau \int_0^{T-W} \frac{u}{\tau} e^{-u/\tau} \frac{du}{\tau}$$

$$= (T-W) \left(1 - e^{-\frac{T-W}{\tau}}\right) - \tau \left(-\frac{T-W}{\tau} e^{-\frac{T-W}{\tau}} + 1 - e^{-\frac{T-W}{\tau}} \right)$$

$$E = (T-W) - \tau \left(1 - e^{-\frac{T-W}{\tau}}\right)$$

or, with $\alpha = \frac{T-W}{\tau}$

$$E = \tau(\alpha - 1 + e^{-\alpha})$$

5

From this, the error probability can be deduced as:

$$10 \quad \text{Error} = \frac{E}{T} = \frac{\tau}{T} (\alpha - 1 + e^{-\alpha})$$

15 Easier-to-handle bounds can be calculated by letting: $F(\alpha) = \frac{\alpha^3}{6} - \frac{\alpha^2}{2} + \alpha - 1 + e^{-\alpha}$

Its derivatives are: $F'(\alpha) = \frac{\alpha^2}{2} - \alpha + 1 - e^{-\alpha}$

20

$$F''(\alpha) = \alpha - 1 + e^{-\alpha}$$

$$F'''(\alpha) = 1 - e^{-\alpha}$$

Since $F(0) = F'(0) = F''(0) = 0$, and $F'''(\alpha)$ is positive for any $\alpha \geq 0$, $F'(\alpha)$, $F''(\alpha)$ and, ultimately, $F(\alpha)$ are as well. So:

$$\frac{\alpha^2}{2} - \frac{\alpha^2}{6} \leq \alpha - 1 + e^{-\alpha} \leq \frac{\alpha^2}{2}$$

25

WU 0207364

PCT/711.01/00651

19

There are better, easy-to-find bounds for the case of $u \geq 1$, but that is not an interesting case ($T \cdot W \gg \tau$). Hence:

$$\begin{aligned} 5 \quad & r \left(\frac{\alpha^2}{2} - \frac{\alpha^3}{6} \right) \leq E \leq r \frac{\alpha^2}{2} \\ 10 \quad & \frac{r}{T} \left(\frac{\alpha^2}{2} - \frac{\alpha^3}{6} \right) \leq \rho_m \leq \frac{r}{T} \frac{\alpha^2}{2} \end{aligned}$$

WU 0207364

PCT/JP01/00651

20

Maximal TTL to ensure a given error probability

An upper bound should be set on TTL which still ensures that the error probability is below a given threshold p_b .

5

$$P_{err} \leq p_0 \Leftrightarrow \frac{\tau}{T} (\alpha - 1 + e^{-\alpha}) \leq p_b$$

10

$$\Leftrightarrow \frac{\tau}{T} \frac{\alpha^2}{2} \leq p_b$$

15

$$\Leftrightarrow \frac{(T - W)^2}{\tau} \leq 2 p_b T$$

$$\Leftrightarrow T^2 - 2(W + \eta_b)T + W^2 \leq 0$$

20

$$\Leftrightarrow W + \tau p_b - \sqrt{(W + \tau p_b)^2 - W^2} \leq T \leq W + \tau p_b + \sqrt{(W + \tau p_b)^2 - W^2}$$

$$\Leftrightarrow W \leq T \leq (W + \tau p_b) \left(1 + \sqrt{1 - \left(\frac{W}{W + \tau p_b} \right)^2} \right)$$

WU 02/07364

PCT/JP01/00651

21

Therefore, an acceptable TTL is:

$$T = (W + \tau p_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right\}$$

5

Maximal TTL, to ensure a given error rate

10 This is the same as for the error probability, replacing p_0 by n_0/n as:

$$n_{err} = np_{err}$$

15

$$n_{err} \leq n_0 \Leftrightarrow p_{err} \leq \frac{n_0}{n}$$

$$\Leftrightarrow T \leq \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

20

25 Therefore, an acceptable TTL is:

$$T = \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

30

35 Average delay for a given TTL

It is useful to consider the average delay D between the cached copy and the original object in the case where the latter has changed during a given interval. First, calculate $\langle \Delta t(u) \rangle$ which is the average delay knowing that the change occurred at a time u between 0 and T .

40

$$\begin{aligned} \langle \Delta t(u) \rangle &= \frac{1}{T} \int_0^T dv (v - u) \\ &= \frac{(T - u)^2}{2T} \end{aligned}$$

45

WJ 0207364

PCT/11.01/0651

22

From this, the overall delay $\langle \Delta t \rangle$ can be deduced by integrating $\langle \Delta t(u) \rangle$ times the probability that a change occurred between u and $u + du$, over $[0, T]$:

$$\begin{aligned}
 \langle \Delta t \rangle &= \int_0^T e^{-r'u} \cdot \frac{du}{r} \cdot \frac{(T-u)^2}{2T} \\
 &= \frac{1}{2T} e^{-r'T} \int_0^T e^{\frac{r}{2}u} \frac{du}{r} (u-T)^2 \\
 &= \frac{r^2}{2T} e^{-r'T} \int_0^T x^2 e^{-x} dx \\
 &= \frac{r^2}{2T} e^{-r'T} \left\{ \left(\frac{T}{r} \right)^2 e^{\frac{r}{2}T} + \int_0^T 2x e^{-x} dx \right\} \\
 &= \frac{r^2}{2T} e^{-r'T} \left\{ \left(\frac{T}{r} \right)^2 e^{\frac{r}{2}T} - 2 \frac{T}{r} e^{\frac{r}{2}T} + 2 \int_0^T e^{-x} dx \right\} \\
 &= \frac{r^2}{2T} e^{-r'T} \left\{ \left(\frac{T}{r} \right)^2 - 2 \frac{T}{r} e^{\frac{r}{2}T} (1 - e^{-\frac{r}{2}T}) \right\} \\
 &= \frac{r^2}{2T} \left\{ \left(\frac{T}{r} \right)^2 - 2 \frac{T}{r} + 2 \left(e^{-\frac{r}{2}T} - 1 \right) \right\} \\
 \langle \Delta t \rangle &= \frac{r^2}{T} \left\{ \frac{1}{2} \left(\frac{T}{r} \right)^2 - \frac{T}{r} + 1 - e^{-\frac{r}{2}T} \right\}
 \end{aligned}$$

25 Since the $\langle \Delta t \rangle$ calculation does not take into account that there has been a change within $[0, T]$, it should be divided by the corresponding probability to get D :

$$\begin{aligned}
 D &= \frac{\langle \Delta t \rangle}{1 - e^{-r'T}} \\
 &= \frac{1}{1 - e^{-r'T}} \cdot \frac{r^2}{T} \left\{ \frac{1}{2} \left(\frac{T}{r} \right)^2 - \frac{T}{r} + 1 - e^{-\frac{r}{2}T} \right\}
 \end{aligned}$$

30

WU 0207364

PC/T/01/00651

23

and since: $\frac{\alpha^3}{6} - \frac{\alpha^2}{2} + \alpha - 1 + e^{-\alpha} \geq 0$

and: $\frac{\alpha^2}{2} - \alpha + 1 - e^{-\alpha} \geq 0$

then:

$$D \leq \frac{1}{\frac{T}{r} - \frac{1}{2} \left(\frac{T}{r} \right)^2} \frac{r^2}{T} \frac{1}{6} \left(\frac{T}{r} \right)^3$$

$$\leq \frac{1}{1 - \frac{T}{2r}} \frac{T}{6}$$

15

In order to limit D to D_0 , a TTL should be chosen, such as:

$$D \leq D_0 \Rightarrow \frac{1}{1 - \frac{T}{2r}} \frac{T}{6} \leq D_0$$

$$\Rightarrow \frac{T}{6} \leq D_0 - D_0 \frac{T}{2r}$$

$$\Rightarrow \left(\frac{1}{6} + \frac{D_0}{2r} \right) T \leq D_0$$

$$\Rightarrow T \leq \frac{6D_0}{1 + 3 \frac{D_0}{r}}$$

30

WU 0107364

PC1711.01/00651

24

From the above, the following tables of TTL ($T(s)$) versus a number of error probabilities, for various values of τ and FP can be derived:

$FP = 15s, \tau = 30s$			
$p_a(\%)$	$T(s)$	$p_m(\%)$	$D(s)$
0.1	15	0	3.3
0.5	17	0.4	4.0
1	18	0.8	4.3
5	23	4.3	6.2
10	27	7.8	8.2

$FP = 15s, \tau = 60s$			
$p_a(\%)$	$T(s)$	$p_m(\%)$	$D(s)$
0.1	16	0.1	3.1
0.5	18	0.4	3.5
1	19	0.7	3.8
5	27	4.2	5.8
10	35	8.5	8.2

$FP = 15s, \tau = 600s$			
$p_a(\%)$	$T(s)$	$p_m(\%)$	$D(s)$
0.1	19	0.1	3.2
0.5	27	0.4	4.6
1	35	0.9	6.0
5	87	4.8	15.6
10	148	9.3	28.1

$FP = 15s, \tau = 3600s$			
$p_a(\%)$	$T(s)$	$p_m(\%)$	$D(s)$
0.1	29	0.1	4.9
0.5	62	0.5	10.4
1	99	1.0	16.7
5	389	4.8	68.5
10	749	9.3	139.3

WU 02/07364

PC/T/11.01/00651

25

$W = 7s, \tau = 60s$			
$P_e(\%)$	$T(s)$	$P_{er}(\%)$	$D(s)$
0.1	7	0	1.2
0.5	9	0.4	1.6
1	10	0.7	1.8
5	17	4.6	3.3
10	23	8.5	4.7

$W = 7s, \tau = 600s$			
$P_e(\%)$	$T(s)$	$P_{er}(\%)$	$D(s)$
0.1	10	0.1	1.7
0.5	17	0.5	2.9
1	23	0.9	3.9
5	73	4.8	13.0
10	133	9.3	24.9

From the above, the following conclusions can generally be made.

- 10 The probability of the retrieved information being "stale" (older than W) is essentially zero for values of $T(s)$ (Time to live) less than or equal to W and increases with increasing $T(s)$ according to a decaying exponential, approaching 100% probability of error at infinity. This observation that the error probability is zero for values of $T(s)$ less than or equal to W is essentially a "trivial" result, since it is clear that no information can be older than W if it is
- 15 updated more frequently than W . Clearly, however, the average update interval τ has a significant effect on how steeply the error probability climbs for values of $T(s)$ greater than W . The greater the average update interval τ with respect to W , the less sharply the error probability rises with increasing values of $T(s)$.
- 20 Figure 4 is a graph 400 illustrating the evolution of error probability P_{er} (vertical axis, expressed as a percentage (%)), as a function of the TTL (in seconds (s)) for various values of τ ($W = 15s$). The line 402 is for $\tau = 20s$, the line 404 is for $\tau = 60s$, the line 406 is for $\tau = 300s$, and the line 408 is for $\tau = 600s$.
- 25 A graphical illustration of a technique for choosing a TTL to maintain error probability below a threshold value P_{th} is obtained by identifying the value of P_{th} on the vertical axis of the graph 400 and following an imaginary line horizontally across the graph 400 to where it intersects the curve (402, 404, 406, or 408) for the appropriate value of τ . By way of example, if an error probability threshold value of 10% is chosen, then the TTL for $\tau = 20s$ (line 402) is a little under 30 seconds, for $\tau = 60s$ (line 404) it is about 35 seconds, for $\tau = 300s$ (line 406) it is about 85 seconds, and for $\tau = 600s$ (line 408) it is about 150 seconds.
- 30

WU 0207364

PCT/JP01/00651

26

By deploying caches for static and dynamic content at appropriate locations in the Internet, and by selecting appropriate update characteristics for cached dynamic content as described hereinabove, effective user response times to both dynamic and static content can be reduced while simultaneously reducing congestion on the external network (i.e., the Internet).

5 The first time an end user (110) receives a web page, the web page may optionally plant a cookie in the user's browser. Thereafter, every time the end user accesses the web page, the browser sends the cookie along with the access request. The cookie tells the server (104) what the server wants (or needs) to know about the end user, in addition to simply the fact
10 that the user wants to retrieve the web page. For example, if the web page is a personalized web page of a single end user, the server knows the end user's preferences.

According to a feature of the invention, the cache (108) can look at the cookie and see if the requested web page is a personalized page of a single end user, or if the cookie indicates the
15 access request is coming from a global end user. If the requested web page is a personalized web page for a single end user (or if there is some other indication that the web page is supposed to be modified each time that it is accessed), then the web page should not be cached. This can be accomplished by setting TTL = 0. Else (e.g., global end user), TTL is simply set as described hereinabove.

20 Although the invention has been described with respect to a limited number of embodiments, it will be appreciated that many variations, modifications and other applications of the invention may be made, and are intended to be within the scope of the invention, as disclosed herein.
25

WU 0207364

PCT/JP01/00651

27

CLAIMS

What is claimed is:

1. System for caching objects having dynamic content on the Internet, comprising:
a cache connected in the Internet for storing and updating copies of dynamic content.
2. System, according to claim 1, wherein:
the cache is disposed at a location selected from the group consisting of major Internet switching locations, dial-in aggregation points, and corporate gateways.
3. System, according to claim 1, wherein:
the cache also stores and updates copies of static content.
4. System, according to claim 1, further comprising:
means for monitoring one or more of the objects to determine update characteristics thereof; and
means for adjusting a time to live (TTL) parameter for the objects based upon the update characteristics.
5. System, according to claim 4, further comprising:
means for determining an age of an object; and
means for updating the object if the TTL for the object is less than its age.
6. System, according to claim 5, wherein:
the TTL for each object is calculated according to an average time of changes for the object.
7. System, according to claim 4, further comprising:
means for determining a probability of error for each of the objects; and
means for adjusting the TTL for each of the objects to maintain its probability of error below a predetermined error probability threshold.
8. System, according to claim 7, wherein:
the TTL for each object is calculated according to the equation:

$$T \leq (W + \tau p_0) \left[1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right]$$

wherein:

- T is the Time To Live, for the dynamic object;
 W is a given time since the original object has changed (i.e., how long it is outdated);
 τ is an average time between changes, which is determined from the object's change statistics; and
 p_0 is maximum error probability, which is the average ratio of the number of requests fulfilled using a cached copy whose corresponding original object has changed for more than the given time W , over the total number of requests.

9. System, according to claim 4, further comprising:
 means for determining an error rate for each of the objects; and
 means for adjusting TTL for each of the objects to maintain its error rate below a predetermined error probability threshold.

10. System according to claim 9, wherein:
 the TTL for each object is calculated according to the equation:

$$T \leq \left(W + \tau \frac{n_0}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_0}{n}} \right)^2} \right\}$$

wherein:

T is the Time To Live for the dynamic object;
 W is a given time since the original object has changed (i.e., how long it is outdated);
 τ is an average time between changes, which is determined from the object's change statistics; and
 n is number of user requests per time unit (e.g., frequency); and
 n_0 is maximum error rate, which is the average number per time unit of requests fulfilled using a cached copy whose corresponding original object has changed for more than the given time W .

11. System, according to claim 4, further comprising:
 means for determining a delay time for each of the objects; and
 means for adjusting TTL for each of the objects to maintain its delay time below a predetermined delay threshold.

12. System according to claim 11, wherein:
 the TTL for each object is calculated according to the equation:

$$T \leq \frac{6D_0}{1 + 3 \frac{D_0}{\tau}}$$

wherein:

T is the Time To Live for the dynamic object;
 τ is an average time between changes, which is determined from the object's change statistics; and
 D_0 is maximum delay, which is the average time between an object change and when the cached copy is refreshed.

13. System, according to claim 4, further comprising:
 means for determining at least one object characteristic selected from the group consisting of error probability, error rate and delay time for each of the objects; and
 means for adjusting TTL to maintain the selected object characteristics below a respective threshold value.

WU 02/07364

PCT/11/01/00651

29

14. System, according to claim 13, further comprising:
means for limiting adjustment of TTL for each of the objects to a range bounded by predetermined minimum (T_{min}) and maximum (T_{max}) values for TTL.
15. Method of responding to a user request for information having dynamic content, comprising:
storing a copy of the dynamic content in a cache;
establishing a time to live (TTL) for the dynamic content;
receiving the user request at the cache;
responding to the user request with the stored copy of the dynamic content if its TTL is greater than its age; and
retrieving an updated copy of the dynamic content and responding to the user request with the updated copy if the TTL of the stored copy is less than its age.
16. Method, according to claim 15, further comprising:
storing a copy of static content in the cache.
17. Method, according to claim 15, further comprising:
determining an average update frequency for the dynamic content; and
determining the TTL for the dynamic content as a function of its average update frequency.
18. Method, according to claim 15, further comprising:
determining an average update frequency for the dynamic content; and
determining the TTL for the dynamic content as a function of its average update frequency and a predetermined error probability threshold.
19. Method, according to claim 18, further comprising:
adjusting the TTL for the dynamic content according to a frequency of user requests for the dynamic content.
20. Method, according to claim 15, wherein:
the TTL for each object is calculated according to the equation:

$$T \leq (W + \tau p_0) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau p_0} \right)^2} \right\}$$

wherein:

- T is the Time To Live, for the dynamic object;
 W is a given time since the original object has changed (i.e., how long it is outdated);
 τ is an average time between changes, which is determined from the object's change statistics; and
 p_0 is maximum error probability, which is the average ratio of the number of requests fulfilled using a cached copy whose corresponding original object has changed for more than the given time W , over the total number of requests.

21. Method, according to claim 15, wherein:

WU 0207364

PCT/JP00/00651

30

the TTL for each object is calculated according to the equation:

$$T \leq \left(W + \tau \frac{n_e}{n} \right) \left\{ 1 + \sqrt{1 - \left(\frac{W}{W + \tau \frac{n_e}{n}} \right)^2} \right\}$$

wherein:

- T is the Time To Live for the dynamic object;
- W is a given time since the original object has changed (i.e., how long it is outdated);
- τ is an average time between changes, which is determined from the object's change statistics;
- n is number of user requests per time unit (e.g., frequency); and
- n_e is maximum error rate, which is the average number per time unit of requests fulfilled using a cached copy whose corresponding original object has changed for more than the given time W .

22. Method, according to claim 15, wherein:
the TTL for each object is calculated according to the equation:

$$T \leq \frac{6D_0}{1 + 3 \cdot \frac{D_0}{\tau}}$$

wherein:

- T is the Time To Live for the dynamic object;
- τ is an average time between changes, which is determined from the object's change statistics; and
- D_0 is maximum delay, which is the average time between an object change and when the cached copy is refreshed.

23. Method, according to claim 15, wherein the information is represented as a web page, the method further comprising:
a first time the user receives the web page, the web page plants a cookie in the user's browser.

24. Method, according to claim 23, wherein:
every subsequent time the end user requests the web page, the browser sends the cookie.

25. Method, according to claim 24, further comprising:
the cache looks at the cookie and see if the requested web page is a personalized page of a single end user.

26. Method, according to claim 25, further comprising:

WO 02/07364

PCT/JP01/00651

31

if the requested web page is a personalized web page for a single end user, then the web page is not cached.

27. Method, according to claim 15, further comprising:
if the information is supposed to be modified each time it is accessed, setting TTL = 0.
28. Method of responding to a user request for an object having dynamic content, said object originating from a server, comprising:
storing a copy of the object in a cache;
establishing a time to live (TTL) for the object;
receiving the user request at the cache;
fulfilling the user request with the stored copy of the object if its TTL is greater than its age; and
fetching an updated copy of the object and responding to the user request with the updated copy if the TTL of the stored copy is less than its age.
29. Method, according to claim 28, further comprising, in the cache:
first setting the TTL for the object to a reasonable lower limit (Tmin); and
adjusting the TTL for the object based on the frequency at which the object actually changes.
30. Method, according to claim 28, further comprising:
each time the cache fetches the object from the server, performing the following procedures:
a. if another fetch for the same object is ongoing, waiting for the previous fetch to complete;
b. fetching the object from the server;
c. replacing the cached copy, if present, by the fetched object, after having compared them to determine whether the object had changed since it was last fetched;
d. initializing or updating the object's change statistics accordingly;
e. marking the object as static or dynamic content depending on the server's reply; and
f. if the object is dynamic, setting its TTL (T) to an appropriate value with respect to an average time between changes (t) determined from the object's change statistics, the number of user requests per time unit (n) determined from the object's access statistics, and one of the following procedures (A-E):
A. maximum error probability;
B. maximum error rate;
C. maximum delay;
D. any combination of the above (A,B,C), taking the lowest T; or
E. any combination of the above (A,B,C), but keeping T within a predetermined window of "reasonable" values bounded by Tmin and Tmax.
31. Method, according to claim 28, wherein the information is represented as a web page, the method further comprising:
a first time the user receives the web page, the web page plants a cookie in the user's browser.
32. Method, according to claim 31, wherein:

WU 02/07364

PCT/JP01/00651

32

every subsequent time the end user requests the web page, the browser sends the cookie.

33. Method, according to claim 32, further comprising:
the cache looks at the cookie and see if the requested web page is a personalized page of a single end user.
34. Method, according to claim 31, further comprising:
if the requested web page is a personalized web page for a single end user, then the web page is not cached.
35. Method, according to claim 28, further comprising:
if the information is supposed to be modified each time it is accessed, setting TTL = 0.

W1 02/07364

PCT/IL01/00651

1/2

Figure 1

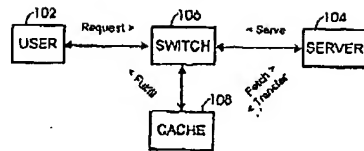
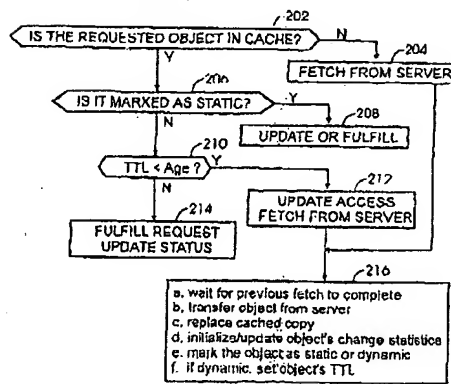


Figure 2



WU 0207364

PCT/JP01/00651

2/2

Figure 3

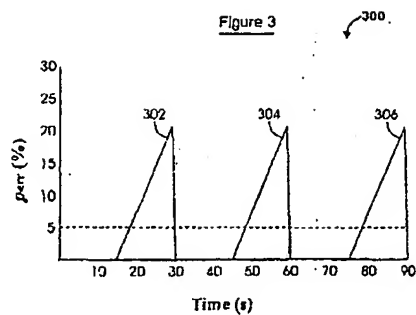
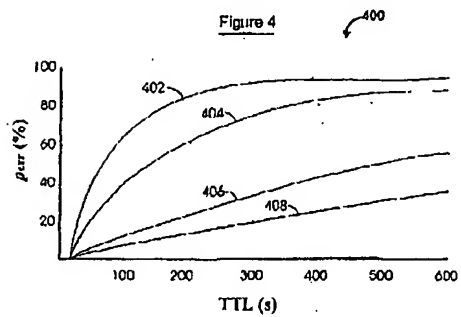


Figure 4



【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

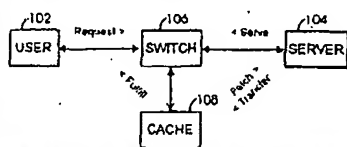
119) World Intellectual Property Organization
International Bureau(43) International Publication Date
24 January 2003 (24.01.2002)

PCT

(10) International Publication Number
WO 02/07364 A3

- (51) International Patent Classification: C06F 13/00
- (11) International Application Number: PCT/IL01/00051
- (22) International Filing Date: 16 July 2001 (16.07.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Date: 17 July 2000 (17.07.2000) US
60218559
- (71) Applicant (for all designated States except US): EPLINK-
ATION NETWORKS LTD. (IL); 58 Haharon St., 45201
Hod Hasharon (IL)
- (72) Inventor and
(73) Inventor Applicants (for US only): MELAMED,
Shmuel (IL); 53 Zaria St., 45350 Hod Hasharon (IL);
BRIEN, Yosef (IL); 12 Barzani St., 69121 Tel Aviv
(IL)
- (74) Agent: FRIEDMAN, Mark, M.; 80 Samachoff, 7 Han-
nam St., 67897 Tel Aviv (IL)
- (84) Designated States (national): AF, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR,
CZ, DE, DK, DM, DZ, EC, EE, FI, FR, GB, GR, HU,
HR, IL, ID, IE, IN, IS, JP, KE, KG, KH, KR, KZ, LC, LK,
LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX,
MY, NZ, NO, NZ, PT, RU, SD, SE, SG, SI, SK, SL,
TH, TJ, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM,
KE, LS, MW, SZ, SD, SL, SZ, TZ, UG, ZW); Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM); European
patent (AT, BE, CH, CY, DE, DK, ES, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE, TR); OAPI patent (BF, BJ, CG,
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
with international search report
- (85) Date of publication of the international search report:
2 May 2002
- For two-letter codes and other abbreviations, refer to the "Table
and Symbols and Abbreviations" appearing at the beginning
of each regular issue of the PCT Gazette.

(54) Title: BANDWIDTH SAVINGS AND QOS IMPROVEMENT IN WWW SITES BY CATCHING STATIC AND DYNAMIC CONTENT ON A DISTRIBUTED NETWORK OF CACHES



(57) Abstract: Caches (108) are dispersed in the Internet for saving and updating copies of objects having dynamic contents. Update characteristics of the objects are determined, and a time to live (TTL) parameter for the objects is adjusted based upon the update characteristics. Generally, the object is updated if its TTL is less than its age. The TTL for an object may be adjusted to (i) maintain its probability of error below a predetermined error probability threshold; (ii) maintain its error rate below a predetermined error probability threshold; or (iii) maintain its delay time below a predetermined delay threshold. Preferably, the caches (108) are dedicated machines and are placed so that web browsing passes through the cache instead of going all the way to the original sites (104), in many different locations, ideally within the network of ISPs (106) providing the Internet connectivity to the highest number of users (102) in those locations. In this manner, the users (102) of those ISPs (106) and, to a lesser extent, neighboring ISPs, will enjoy a huge QoS and speed improvement, for most of the traffic will stay within or close to the ISP's (106) internal networks and not need to go through the highly-loaded Internet backbone; and the original web sites (104) will no longer need as much bandwidth, since the caches (108) will absorb most of the load. The system can adapt, in real time, according to the number of requests to each page and the actual update frequency of the page.

WO 02/07364 A3

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MC,MK,MN,MW,MX,MZ,NO,NZ,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TR,TT,TZ,UA,UG,US,UZ,VN,YU,ZA,ZW

(72)発明者 シュムエル メラメド

イスラエル国、ホド ハシャロン 45350、ザミル ストリート 53

(72)発明者 イベス ビギオ

イスラエル国、テル アビブ 69121、バラザニ ストリート 12

Fターム(参考) 5B056 BB00 BB62 BB64 HH01

5B082 FA12

【要約の続き】

、そして狭義には、隣接するISPのユーザは、巨大なQoSとスピードの改善とを享受する。また、キャッシュが負荷の大部分を吸収するため、本来のウェブ・サイトは、もはや高いビット・レートが必要としない。このシステムは、各ページに要求数及び実際のページ更新頻度に応じて、リアルタイムで適応することができる。

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.